

(E)

(19) 日本国特許庁 (J P)

(12) 公表特許公報 (A)

(11) 特許出願公表番号

特表2003-528358

(P2003-528358A)

(43) 公表日 平成15年9月24日 (2003.9.24)

(51) Int.Cl. <sup>7</sup>	識別記号	F I	テ-マ-ト (参考)
G 0 6 F 17/60	1 6 2	G 0 6 F 17/60	1 6 2 C 5 B 0 8 5
13/00	6 3 0	13/00	6 3 0 G
15/00	3 9 0	15/00	3 9 0

審査請求 未請求 予備審査請求 有 (全117頁)

(21) 出願番号 特願2000-569298(P2000-569298)  
 (86) (22) 出願日 平成11年8月24日 (1999.8.24)  
 (85) 翻訳文提出日 平成13年2月26日 (2001.2.26)  
 (86) 国際出願番号 P C T / U S 9 9 / 1 9 2 3 2  
 (87) 国際公開番号 W O 0 0 / 0 1 4 6 1 8  
 (87) 国際公開日 平成12年3月16日 (2000.3.16)  
 (31) 優先権主張番号 6 0 / 0 9 7 , 5 8 3  
 (32) 優先日 平成10年8月24日 (1998.8.24)  
 (33) 優先権主張国 米国 (U S)  
 (31) 優先権主張番号 6 0 / 0 9 7 , 8 7 9  
 (32) 優先日 平成10年8月25日 (1998.8.25)  
 (33) 優先権主張国 米国 (U S)

(71) 出願人 富士通株式会社  
 神奈川県川崎市中原区上小田中4丁目1番1号  
 (72) 発明者 アデル ゴウニイミー  
 アメリカ合衆国, カリフォルニア州  
 94010, ヒルズバロウ, フォレスト ビュー  
 アベニュー 2325  
 (72) 発明者 アリ ジナトバクシ  
 アメリカ合衆国, カリフォルニア州  
 95148, サンホセ, ハンバーサイド コー  
 ト 3138  
 (74) 代理人 弁理士 酒井 宏明

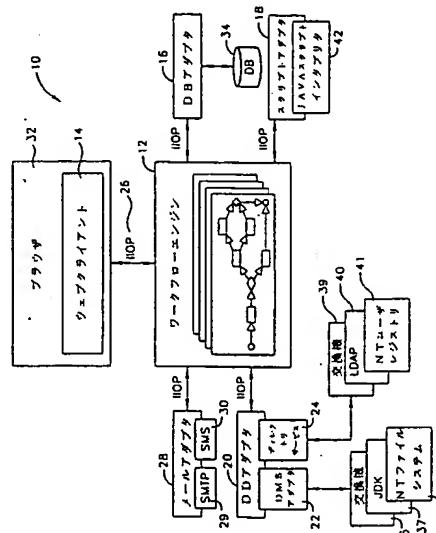
最終頁に続く

(54) 【発明の名称】 ワークフローシステムおよび方法

(57) 【要約】

【課題】 作業プロセスを自動化するワークフローシステムであり、本発明の所望される結果とは、相対的に容易な配備および高められた柔軟性を提供する。

【解決手段】 本発明にかかるワークフローシステムは、多岐に渡るクライアントおよびアダプタを使用する。開発者は、比較的容易にアプリケーションを立ち上げ、実行するためにワークフローシステムによって提供されたクライアントを使用することができる。該ワークフローシステムは、迅速な (DMS)、(DS)、および (DB) 配備および要素の柔軟性を可能にするアダプタを提供する。



## 【特許請求の範囲】

【請求項1】 ワークフロープロセスを生成、制御するためにプログラムされたコンピュータシステムであって、

ワークフロープロセスの定義および実施のために第1コンピュータをベースにしたシステムにより実行されるワークフローエンジンと、

前記ワークフローエンジンに結合される第2コンピュータをベースにしたシステムによって実行されるクライアントと、

1台または複数台のサーバと、

前記ワークフローエンジンと一つまたは複数のサービスの内の前記一つの間に通信機構を提供するためのアダプタであって、オブジェクト指向型であり、プラットフォームに依存しない、言語に依存しないインタフェースである固定インタフェースを有し、前記ワークフローエンジンが共通インタフェースに通信できるようにし、それにより前記ワークフローエンジンをサービスから独立させ、前記ワークフローエンジンに対するコード変更なしに、サービスを変更できるようにする前記アダプタと、

を備えることを特徴とするコンピュータシステム。

【請求項2】 前記クライアントが、ウェブブラウザ内で実行することを特徴とする請求項1に記載のプログラムされたコンピュータシステム。

【請求項3】 前記ワークフローエンジンが、J A V A（登録商標）プログラミング言語を使用して実現されることを特徴とする請求項1に記載のプログラムされたコンピュータシステム。

【請求項4】 一つまたは複数のサービスの内の一つがデータベース管理システムであり、前記アダプタが、前記ワークフローエンジンと前記データベース管理システム間に通信機構を提供することを特徴とする請求項1に記載のプログラムされたコンピュータシステム。

【請求項5】 一つまたは複数のサービスの内の一つが文書管理システムであり、前記アダプタが前記ワークフローエンジンと前記文書管理システム間に通信機構を提供することを特徴とする請求項1に記載のプログラムされたコンピュータシステム。

【請求項6】 前記アダプタ機能が、指定されたユーザ向けの文書管理サービスオブジェクトを作成することと、文書管理サービスオブジェクトを破壊することと、を含むことを特徴とする請求項5に記載のプログラムされたコンピュータシステム。

【請求項7】 前記アダプタ機能が、さらに、指定された経路用の文書／ディレクトリのリストを戻すことと、読取りまたは書込み用に指定されたファイルを貸し出すことと、前記指定されたファイルを返すことと、を含むことを特徴とする請求項6に記載のプログラムされたコンピュータシステム。

【請求項8】 前記アダプタ機能が、さらに、ファイルが読取りモードのために貸し出される添付物の経路を戻すことと、ファイルが書込みモードのために貸し出される添付物の経路を戻すことと、を含むことを特徴とする請求項6に記載のプログラムされたコンピュータシステム。

【請求項9】 一つまたは複数のサービスの一つがディレクトリサービスであり、前記アダプタが、前記ワークフローエンジンと前記ディレクトリサービス間に通信機構を提供することを特徴とする請求項1に記載のプログラムされたコンピュータシステム。

【請求項10】 前記アダプタ機能が、さらに、指定されたユーザ向けのディレクトリセッションオブジェクトを作成することと、ディレクトリセッションオブジェクトを破壊することと、を含むことを特徴とする請求項9に記載のプログラムされたコンピュータシステム。

【請求項11】 前記アダプタ機能が、さらに、指定されたグループ向けのユーザのリストを戻すことと、一つの長い文字列として、指定されたグループ内のユーザのリストを戻すことと、グループのリストを戻すことと、を含むことを特徴とする請求項10に記載のプログラムされたコンピュータシステム。

【請求項12】 前記アダプタ機能が、さらに、指定されたユーザ用のすべてのプロパティを戻すことと、前記指定されたユーザ用のプロパティを作成することと、前記指定されたユーザ用のプロパティを更新することと、前記指定されたユーザのプロパティを削除することと、を含むことを特徴とする請求項10に記載のプログラムされたコンピュータシステム。

【請求項13】 第2サービスが文書管理システムであり、第2アダプタが前記ワークフローエンジンと前記文書管理システム間に通信機構を提供することを特徴とする請求項9に記載のプログラムされたコンピュータシステム。

【請求項14】 さらに、文書管理システムオブジェクトを前記文書管理システムから取得するため、およびディレクトリを前記ディレクトリサービスから取得するため、ならびに前記オブジェクトを前記ワークフローエンジンに提供するためのDocument Directory（文書ディレクトリ）アダプタを備えることを特徴とする請求項13に記載のプログラムされたコンピュータシステム。

【請求項15】 前記アダプタが、前記文書管理システムが実行中である機械で利用できるすべてのドライブを戻すことを特徴とする請求項5に記載のプログラムされたコンピュータシステム。

【請求項16】 一つまたは複数のサービスの一つがスクリプトインタプリタであり、前記アダプタが前記ワークフローエンジンと前記スクリプトインタプリタ間に通信機構を提供することを特徴とする請求項1に記載のプログラムされたコンピュータシステム。

【請求項17】 一つまたは複数のサービスの一つが電子メールサービスであり、前記アダプタが前記ワークフローエンジンと前記電子メールシステム間に通信機構を提供することを特徴とする請求項1に記載のプログラムされたコンピュータシステム。

【請求項18】 一つまたは複数のサービスの一つが、クライアントが、前記クライアントが、それが第1コンピュータをベースにしたシステム上で存在するときに前記オブジェクトに対するポインタを参照する必要なく、前記オブジェクトの命名サービス名を参照することによって、第1コンピュータをベースにしたシステムによって実行されるオブジェクトを参照することができ、それによってオブジェクトが第1コンピュータをベースにしたシステム内の別のロケーションに移動されるにしても、クライアントがオブジェクトを参照できる、命名サービスであることを特徴とする請求項1に記載のプログラムされたコンピュータシステム。



【請求項19】 第1コンピュータをベースにしたシステムが、複数の接続されたコンピュータを備えることを特徴とする請求項18に記載のプログラムされたコンピュータシステム。

【請求項20】 前記ワークフローエンジンと前記クライアント間のインタフェースが、共通オブジェクトリクエストブローカーアーキテクチャインタフェースによって支配されることを特徴とする請求項1に記載のプログラムされたコンピュータシステム。

【請求項21】 前記ワークフローエンジンと前記クライアント間のインタフェースが、インターネットORB間プロトコルによって支配されることを特徴とする請求項1に記載のプログラムされたコンピュータシステム。

【請求項22】 前記ワークフローエンジンと前記アダプタ間のインタフェースが、共通オブジェクトリクエストブローカーアーキテクチャインタフェースによって支配されることを特徴とする請求項1に記載のプログラムされたコンピュータシステム。

【請求項23】 前記ワークフローエンジンと前記アダプタ間のインタフェースが、前記インターネットORB間プロトコルによって支配されることを特徴とする請求項1に記載のプログラムされたコンピュータシステム。

【請求項24】 前記データベースアダプタが、前記ワークフローエンジンと前記データベース管理システム間に通信機構を提供するためにJava DataBase Connectivity (Java DataBase接続性)規格を使用することを特徴とする請求項4に記載のプログラムされたコンピュータシステム。

【請求項25】 さらに、前記クライアントが前記ワークフローエンジンおよび前記アダプタと通信できるようにするためのウェブサーバと、を備えることを特徴とする請求項1に記載のプログラムされたコンピュータシステム。

【請求項26】 クライアントオブジェクトの状態をカプセル化するモデル層と、

クライアントプレゼンテーションのため、およびユーザ入力进行处理のためのユーザインタフェース層と、

を備えることを特徴とするワークフロークライアント。

【請求項27】 前記モデル層が、

前記クライアントのすべての構成要素間でのイベントディスパッチのためのアプリケーション論理と、

ログイン、ログアウト、および前記ワークフローエンジンからのリストの検索のためのワークフローマネージャモデルと、

プロセスインスタンスを処理するためのプロセスモデルと、

書式と前記プロセスインスタンス間のデータ交換のための書式モデルと、

作業アイテムを管理、更新するための作業アイテムモデルと、

計画を処理するためのテンプレートモデルと、

を備えることを特徴とする請求項26に記載のワークフロークライアント。

【請求項28】 前記ユーザインタフェース層が、

プロセス、プロセス定義、および作業アイテムのリストを提示するオーガナイザと、

選択されたプロセスまたはプロセス定義の図表による表記である、縮小サムネイルオーバービューを提示するオーバービューと、

プロセスおよびプロセス定義の表示／編集を可能にするプロセス／テンプレートビューワ／エディタと、

添付物および書式へのアクセスを可能にする活動応答と、

プロセスの詳細なビューを提供する活動詳細ビューと、

ユーザが特定の選択で活動の完了を指定できるようにする応答バーと、

ユーザ定義書式の内容を提示する書式ビューと、

を備えることを特徴とする請求項26に記載のワークフロークライアント。

【請求項29】 前記ユーザインタフェース層が、

メニューバーディスプレイと、

ツールバーディスプレイと、

プロセス、プロセス定義、および作業アイテムのリストのディスプレイと、

選択されたプロセスおよびプロセス定義の図表表記のディスプレイと、

前記プロセスおよび前記プロセス定義を表示、編集するためのディスプレイと

添付物および書式にアクセスするためのディスプレイと、  
ユーザがある特定の選択で活動の完了を指定できるようにするディスプレイと  
  
ユーザ定義書式の内容のディスプレイと、  
を実行することを含むことを特徴とする請求項26に記載のワークフロークライアント。

【請求項30】 前記ディスプレイが、Java Scriptおよびハイパテキストマーク付け言語を使用して統合されることを特徴とする請求項29に記載のワークフロークライアント。

【請求項31】 ユーザが、書式の上に配置されるプロセス属性を指定するための手段と、

ハイパテキストマーク付け言語で前記書式を生成するための手段と、  
を備えることを特徴とするワークフローシステム用の書式ビルダ。

【請求項32】 前記書式を生成するための前記手段が自動であることを特徴とする請求項31に記載の書式ビルダ。

【請求項33】 前記書式が自動的にプロセスに接続されることを特徴とする請求項32に記載の書式ビルダ。

【請求項34】 前記書式が、前記プロセスの内側の活動に自動的に接続されることを特徴とする請求項33に記載の書式ビルダ。

【請求項35】 テンプレート、プロセス、または活動のリストを表示するための領域と、

前記リストから選択されたアイテムの図表表記を表示するための領域と、  
前記リストから選択されたアイテムの情報を表示し、修正するための領域と、  
を備えることを特徴とする画面レイアウトを生成するワークフロープロセスモデルを開発するためのクライアント。

【請求項36】 前記メニューバーが、アプリケーション全体の機能を実行するためであることを特徴とする請求項29に記載のクライアント画面レイアウト。

【請求項37】 Eメールメッセージ内に埋め込まれている全域資源位置指示子(URL)を介して起動されるクライアントであって、

前記タスクに特殊な添付物リストと、

前記タスクに特殊な書式リストと、

割り当てられたタスクの詳細なビューと、

を備えることを特徴とする画面レイアウトを生成するクライアント。

【請求項38】 前記ユーザに割り当てられるタスクの作業リストと、

前記タスクに特殊な添付物リストと、

前記タスクに特殊な書式リストと、

割り当てられたタスクの詳細ビューと、

を備えることを特徴とする画面レイアウトを生成するためのクライアント画面

【請求項39】 テンプレートリストと、

テンプレートの詳細ビューと、

前記テンプレートに関連付けられた添付物リストと、

前記テンプレートに関連付けられた書式リストと、

を備えることを特徴とする画面レイアウトを作成する、プロセスの具体化のためのクライアント。

【請求項40】 コンピュータでの運用のためにクライアントを使用してワークフローを開発するための方法であって、

計画を作成することであって、プロセスの活動を定義すること、前記プロセスの参加者を定義すること、および前記プロセスの属性を定義することを含む前記計画を作成すること、

前記計画をプロセステンプレートとして記憶すること、

前記プロセステンプレートからアクティブプロセスを起動すること、

リアルタイムで前記アクティブプロセスを編集すること、

を含むことを特徴とする方法。

【請求項41】 さらに、添付物を表示することを含み、前記添付物が、前記ユーザが前記添付物を貸し出すときだけに編集することができ、前記編集され

た添付物が、前記編集された添付物が前記ユーザによって返されたときだけ見ることができることを特徴とする請求項40に記載の方法。

【請求項42】 さらに、多くの読取りがあるが、書込みが1度だけである、前記添付物のマスタコピーの回覧を含むことを特徴とする請求項41に記載の方法。

【請求項43】 前記活動、参加者、および前記プロセスの属性が、ポイントおよびクリックの機能性で定義されることを特徴とする請求項40に記載のクライアント。

【請求項44】 前記プロセスの属性が書式内で定義され、それによって前記書式がサードパーティツールを使用してカスタマイズされることを特徴とする請求項40に記載の方法。

【請求項45】 さらに、活動リスト、プロセステンプレートリストおよびアクティブプロセスリストを作成することを含むことを特徴とする請求項40に記載の方法。

【請求項46】 さらに、ユーザによって指定される基準に従って前記リストのどれかを並べ替えることを特徴とする請求項45に記載の方法。

【請求項47】 さらに、ユーザによって指定される基準に従って、前記リストのどれかをフィルタリングすることを特徴とする請求項45に記載の方法。

【請求項48】 さらに、前記活動が前記ワークフローエンジンによって実行される準備が完了するときに、前記活動を表示することを特徴とする請求項40に記載の方法。

【請求項49】 さらに、文書をプロセスと関連付けることを特徴とする請求項40に記載の方法。

【請求項50】 さらに、前記文書を前記プロセスから切り離すことを特徴とする請求項49に記載の方法。

【請求項51】 さらに、活動プロセスの前記履歴を記録し、表示することを特徴とする請求項40に記載の方法。

【請求項52】 ユーザが、コンピュータでの運用のためにクライアントを使用して割り当てられたタスクを処理するための方法であって、

作業リストの作業アイテムにアクセスすること、  
プロセスを具体化すること、  
を含むことを特徴とする方法。

【請求項53】 さらに、ユーザが作業リストからの彼らの割り当てられたタスクに取り組むときに文書を添付、表示、および編集することを特徴とする請求項52に記載の方法。

【請求項54】 さらに、ユーザが作業リストから割り当てられたタスクに取り組むときに書式を表示、編集することを特徴とする請求項52に記載の方法。

【請求項55】 クライアントを使用して、Eメールを通してワークフローエンジンと対話するための方法であって、

全域資源位置指示子(URL)を通して事前に定義されたテンプレートからプロセスを具体化すること、

繋がれていないユーザが、彼らの作業アイテムおよび関連文書を受け取ることができるようにすること、

ユーザが、Eメールを介してそのタスクを完了できるようにすること、  
を含むことを特徴とする方法。

【請求項56】 クライアントを使用するプロセスの具体化のための方法であって、

プロセスを初期化すること、

値を書式の中に記入すること、

添付物を初期化すること、

前記書式に入力された値に属性が初期化されたプロセスの新規インスタンスを作成すること、

を含むことを特徴とする方法。

【請求項57】 ワークフロープロセスモデルを開発するためのクライアントと、

Eメールメッセージ内に埋め込まれているURLを介して起動されるクライアントと、

Eメール接続を通してワークフローエンジンと対話するためのクライアントと、

プロセスの具体化のためのクライアントと、

ユーザが割り当てられたタスクに取り組むためのクライアントと、

を含むことを特徴とする複数のクライアントを有するプログラムされたコンピュータシステム。

【請求項58】 Document Directory（文書ディレクトリ）アダプタがユーザを認証することを特徴とする請求項14に記載のプログラムされたコンピュータシステム。

【請求項59】 Document Directoryアダプタが役割分解を実行することを特徴とする請求項58に記載のプログラムされたコンピュータシステム。

## 【発明の詳細な説明】

## 【0001】

(関連出願に対するクロスリファレンス)

本出願は、後述される「ワークフローシステムおよび方法」と題され、これによりここに完全に述べられるかのように参照して組み込まれる三つの米国仮出願の優先日の恩典を主張する。

1998年8月24日に提出された出願番号第60/097, 583号

1998年8月25日に提出された出願番号第60/097, 879号

1998年8月25日に提出された出願番号第60/097, 791号

1996年2月6日に発行された「ワークプロセス計画をモデリング、解析、および実行するためのシステムおよび方法 (SYSTEM AND METHOD FOR MODELING, ANALYZING AND EXECUTING WORK PROCESS PLANS)」と題されている特許番号第5, 490, 097号、Swensonらは、あたかもここに完全に述べられるかのようにすべての目的のために参照して組み込まれる。

## 【0002】

(発明の分野)

本発明は、概してオープンワークフローアーキテクチャに関し、さらに特定するとワークプロセスを自動化するワークフローシステムに関するものである。

## 【0003】

(発明の背景)

特にプロセスには人間が関係するため、自動化されたプロセスソリューションでは変更はありふれた出来事である。単純な自動化されたプロセスソリューション以上のものを従来のワークフロー製品を使用して作成することは、もっとも経験を積んだ自動化専門家にとってさえ最もやりがいのあるタスクであることが分かっている。長期の複雑な開発サイクルは、複雑な自動化ソリューションにはつきものであり、いったん作成されると、プロセスの進化に伴い修正するのはきわめて困難である場合がある。

## 【0004】



その独占的なシステムおよび旧式のアーキテクチャを用いる現在のワークフロー技術は、典型的なビジネスプロセスを構成する人々、手順およびリソースの変化に対処するのに適さなくなっている。さらに、大部分の既存のワークフロー製品は、技術が進歩する速度についていくことができない。インターネット、情報技術（IT）インフラストラクチャ、およびサポートするワークフロー関連アプリケーションは、連続して進化し、プロセス自動化が長足の進歩を遂げることができるようにするが、ワークフローアプリケーションがこれらの進歩に対処できる場合だけである。その結果、アナリストおよび開発者は、技術の進展の力を利用しつつ、プロセスが進化できるより速く、さらに容易に変化することができるワークフローソリューションを必要とする。

#### 【0005】

ワークフローは、指定されたプロセスのための作業のフローの論理を組み込んだ特殊アプリケーションを開発することによってビジネスプロセスを自動化しようとする作業から進化してきた相対的に新しい概念である。この取り組み方の欠点は、プロセスが変化するときにはいつでも、アプリケーション全体を更新しなければならない、保守が悪夢のようになるという点である。ワークフローは、プロセスのあらゆる面をカプセル化することによってこの問題に対処するために導入された。これには、規則、ルーティング経路、活動、データ等のプロセス定義情報が含まれる。さらに重要なことには、それにはこの情報の管理の自動化が含まれる。そこでワークフローをサポートするアプリケーションは、ワークフロープロセスそれ自体の詳しい知識を必要としなくてもワークフローを実現することができる。

#### 【0006】

ワークフローシステムは、作業プロセスを自動化するために作成された。ワークフローソリューションの歴史は二つの一般的な経路を辿ってきている。すなわち、（1）プリミティブなフレームワーク、および（2）多くの制限がある完全なソリューションである。

#### 【0007】

該プリミティブなフレームワーク方法は、最小の骨組、すなわちワークフロー

システムにとって共通であったが、任意の特定のシステムには適していないコードを提供した。該プリミティブなフレームワークは、それには、ワークフローシステムを任意の特定のアプリケーション用に実行させるために大量のコードを作成することが必要となったため、配備するのが困難であった。

【0008】

他方、該多くの制限がある完全なソリューション方法は、ある特定のアプリケーションの解決を目的とした大量のコードを含んでいる。しかしながら、該多くの制限を持った完全なソリューション方法は、多岐に渡るアプリケーションには適用しなかった。完全なソリューションワークフローシステムは、ある特定の文書管理システム（（DMS））、ディレクトリサービス（DS）およびデータベースシステム（DB）を目的としていた。ワークフローシステムの任意の要素が変更されると、該ワークフローシステムの改定が必要とされた。その結果、（ワークフロー要素に対する更新を含む）ワークフロー要素に変更があったときにはいつでも、継続的な保守が必要とされた。たとえば、該（DMS）が別の（DMS）に変更された場合、それには該ワークフローシステム内のコード変更が必要となった。このようにして、完全なソリューション方法は柔軟性がなく、アプリケーションに対する変更に容易に適応できなかった。

【0009】

（発明の要約）

本発明は、作業プロセスを自動化するワークフローシステムである。本発明の所望される結果とは、相対的に容易な配備および高められた柔軟性である。該ワークフローシステムは、多岐に渡るクライアントおよびアダプタを使用する。開発者は、比較的容易にアプリケーションを立ち上げ、実行するためにワークフローシステムによって提供されたクライアントを使用することができる。該ワークフローシステムは、迅速な（DMS）、（DS）、および（DB）配備および要素の柔軟性を可能にするアダプタを提供する。

【0010】

アダプタは、アプリケーションを（DMS）、（DS）、および（DB）と直接的に結び付けるそれ以外のワークフローシステムとは対照的に、アプリケーション

ョンとは無関係に (DMS)、(DS)、および (DB) を変更できるようにする固定インタフェースを有する。該アダプタによって、開発者は、変化する種類および規模のクライアントを接続できるようにし、開発者がアプリケーション要素を自分達の用途によりよく役立つために混合、整合できるようにする。このようにして、該ワークフローシステムは、(DMS)、(DS) および (DB) が同じベンダ製でなくてもよい、任意の独占的な (DMS)、(DS) または (DM) に結び付けられていない。

#### 【0011】

業界で受け入れられている2種類のワークフロー製品がある。すなわち、Ad HocとProductionである。Ad Hocワークフローは、プロセスに対し、あるにしても、ほとんど制約を課さない構造化されていない自由機式のワークフローとして設計されている。それは、ワードプロセッサ、スプレッドシートプログラム、および電子メールプログラム/システムなどのソフトウェア構成要素を、参加者が共用データ上で作業できるようにリンクしてよい。この種のワークフローは、考えを分かち合う、文書を検討する、および承認のために要求を処理するなどのビジネス機能には有効である。

#### 【0012】

Productionワークフローは、ビジネスプロセスが所定の規則に従っていることを確実にしようとする。すなわち、該プロセスの中の各タスクは、適切な人々が適切なデータにアクセスでき、適切なときにタスクを実行するように命令されるように調整されている。この種のワークフローは、定められた規則に従うことが重要であり、リアルタイムで実行されがちなビジネス機能に有効である。例としては、販売、購入、製造および会計が含まれる。

#### 【0013】

該ワークフローシステムは、これら両方の種類のワークフローをサポートするためにオープンアーキテクチャを使用する。それは、複数のプラットフォームをサポートし、アプリケーションにワークフローエンジンに話をさせることができるアプリケーションプロミシングインタフェース (API) を含み、リレーショナルデータベース管理システム (RDBMS) データベースをサポートし、任意の種

類の作業（形式、画像、実行可能物）のルーティングを可能にする。加えて、その柔軟なアーキテクチャは、ワークフローに対する変更を動的に容易にし、既存のインフラストラクチャとの統合をサポートするように設計されている。

#### 【0014】

（詳細な説明）

以下、この発明にかかるワークフローシステムおよび方法の実施の形態について、図面を参照して詳細に説明する。

#### 【0015】

図1は、総合的な分散ワークフローシステム構成要素アーキテクチャおよび本発明の実施の態様の構成要素対話を示す。該ワークフローシステム10は、分散クライアントサーバ、ウェブでイネーブルされたワークフローシステム／モデルである。さらに、該ワークフローシステムは、協力プロセスおよび管理企業間プロセスを自動化するためのアプリケーション開発ツールである。該ワークフローシステムは、既存のITインフラストラクチャと一体的に統合するカスタマイズ化されたワークフローソリューションを作成するために必要なフレームワーク、ツール、および構成要素を完備したワークフロー開発環境を提供する。これらのソリューションの配備のために、該ワークフローシステムは、ウェブベースインタフェースを通して企業内の一貫した拡張性のあるサポートインフラストラクチャを提供する。ウィンドウをベースにしたその他のワークフローソリューションとは異なり、該ワークフローシステムはウェブをベースにし、該ワークフローシステムがウェブブラウザからアクセス可能であることを意味する。

#### 【0016】

該ワークフローシステムのいくつかの重要な特徴は、以下を含む。すなわち、全種類のプロセスを処理するための企業全体で拡張性のあるインフラストラクチャ、視覚的なビジネスプロセスモデルを事前にまたは実行中に容易に作成、修正する能力（プログラミングは必要とされない）、プロセスステータスの迅速な評価および大きなピクチャの理解のための特徴が豊富なグラフィック表示、編成可能かつフィルタリング可能な汎用実行リスト、割当ての即座の通知、プロセスに関連する文書のロケーションを集中化するための文書管理システムとの統合、ウ

ウェブブラウザからのアクセス性および開発者がワークフローエンジンまたはワークフローイネーブルされる既存の製品と通信するカスタマイズ化されたアプリケーションを作成できるようにするアプリケーションプログラミングインタフェース (API) である。

【0017】

該ワークフローシステムは、ビジネスグループが、協力して、ビジネスプロセスを計画、自動化、追跡調査および改善できるように能力を与えるワークフロー管理フレームワークを提供する。該ワークフローシステムは、該ワークフローのモデルであるだけでなく、該ワークフローシステム/モデルを開発するための開発環境/ツールでもある。該ワークフローシステムは、ワークフロープロセスを開発し、操作する上でエンドユーザに能力を与える。それは、知識作業者が、プログラミングの知識がなくても十分に意味のあるワークフロープロセスを定義できるようにする。該知識作業者は、該ワークフローシステムに追加するまたは該ワークフローシステムを開発するために、ただ1行のコードを書く必要はない。

【0018】

該ワークフローシステムは、実際に任意の既存IT環境にあるワークフローによってイネーブルされたアプリケーションにフレームワークを提供する。そのモジュラー設計およびJAVATM、インターネットORB間プロトコル (IIOP)、ハイパテキストマーク付け言語 (HTML)、およびオブジェクト管理グループ (OMG) およびワークフロー管理連合 (WfMC) プロトコルなどの標準技術の活用により、それは真の相互運用製品となり、ほとんどあらゆるIT環境内での統合に対処する。該サーバアーキテクチャは、ワークフローモデルおよびその要素を拡張するためのメタモデル (複数のモデルのモデル) である。

【0019】

該サーバアーキテクチャは、拡張サーバレベル統合を可能にするオープンアーキテクチャである。追加構成要素は、JavaScriptなどの一般的なスクリプト言語と容易に統合することができる。該アプリケーションプログラミングインタフェースは、クライアントの拡張性および顧客対応のためのJavaBe

an技術を組み込む。そして、一般的なサードパーティのウェブオーサリングツールを活用する能力によって、ユーザは、よく知られている堅牢な開発環境において、ワークフローシステム内で生成される形式をカスタマイズすることができる。

#### 【0020】

ビジネスワークフローに完全な機能性サポートを提供するために、該ワークフローシステム10は、ワークフローエンジン12、クライアント14、Database（データベース）（DB）アダプタ16、Script Adapter（スクリプトアダプタ）18、文書管理システム（（DMS））アダプタ22およびディレクトリサービスアダプタ24を備えるDocument/Directory（文書/ディレクトリ）（DD）Adapter（アダプタ）20を備える。

#### 【0021】

該ワークフローシステムのモジュラーアーキテクチャおよびオープンプラグ接続可能アダプタAPIが、さまざまな文書管理システム、ディレクトリサービス、DataBase（データベース）管理システムおよびサーバ側スクリプトインタプリタを統合する作業を容易にする。クライアントAPIもモジュール化され、コンポーネント化されている。このようにして、Client（クライアント）、Server（サーバ）、およびアダプタ構成要素は、負荷を分散し、それによって該ワークフローシステムの性能を改善するためにさまざまな機械上にインストールされてよい。

#### 【0022】

該ワークフローシステムは、その「アダプタオブジェクト技術」を介して、多くの一般的な文書管理システム、ディレクトリサービス、データベースおよびスクリプト言語と一体的に接続する。「Adapter Object（アダプタオブジェクト）」技術によって、サーバは、必要なITインフラストラクチャ構成要素、すなわち文書管理システム、ディレクトリサービス、およびデータ管理システムと通信できる。スクリプトインタプリタ用のアダプタによっても、共通スクリプト言語を通して追加構成要素との統合が可能になる。これは、新興技術

の統合を可能にし、それにより「ベンダ固定化」から保護することだけではなく、組織がそのITインフラストラクチャに対して行った投資も利用する（「ベンダ固定化」は、ある組織がベンダを切り替えることに関わる多大な費用のためにあるベンダにコミットされるときに発生する）。

#### 【0023】

さらに、該ワークフローシステムは、新しいサポートワークフローアプリケーションまたはインフラストラクチャ製品を管理情報システム（MIS）に対し強制することも必要としない。本発明の好ましい実施の態様においては、該ワークフローシステムは、遠隔サブプロセスおよびワークアイテム通知転送を含む複数のサーバ構成をサポートする。スケーラビリティは、単一サーバあたり少なくとも100人のユーザをサポートするために拡張することができる。

#### 【0024】

該ワークフローシステムの主要な構成要素とは、ワークフローエンジンである。該ワークフローシステムは、独立サービス/ソフトウェアベンダ（ISV）および付加価値小売業者（VAR）に堅牢なワークフローエンジンを提供する。ISVは独自の製品を持ち、専用の製品を機能拡張するためにその他の製品を使用してよい。VARは独自の製品を持たないが、その他の製品に追加し、このようにしてさらに望ましい製品を作る。ワークフロー開発ツールとして、該ワークフローシステムは、それを用いてその既存のアプリケーションを「ワークフローイネーブルし」、それによって新規機能性を作成する完全にフィーチャーされる相互運用エンジンをISVに提供する。さらに、該ワークフローシステムは、VARおよびその他のサードパーティ統合者に、それを用いて特殊化した堅牢なカスタムワークフローソリューションをほぼあらゆるIT環境で使用するために容易に作成するためのツールを与える。

#### 【0025】

好ましい実施の態様においては、該ワークフローエンジンはサーバ上に常駐し、100% Javaをベースにしている。該ワークフローエンジンは、ユーザとその他の構成部品間の対話を交渉し、ユーザによって起動されるプロセスを実行し、ユーザにプロセス内のステータスの変化を通知する。該ワークフローエンジ

ンは、100を超えるリアルタイム（繋がれている）ユーザ、および1000までの「不定期な」（つながれていない）ユーザを、すべて単一システムから、同時に、ウェブを介してサポートする。リアルタイムユーザの場合、記憶および転送は必要とされない。組織が大きくなり、プロセスが進化するにつれて、該ワークフローシステムのモジュラーアーキテクチャは、トランザクションスループットの増加を処理するために、容易に拡大縮小することができる。

【0026】

好ましい実施の態様においては、該Server（サーバ）と該Client（クライアント）の間の情報の転送は、共通オブジェクトリクエストブローカーアーキテクチャ（CORBA）リンク上で発生する。該CORBAインタフェースはプラットフォームに依存しない、言語に依存しないインタフェースであり、CORBAインタフェースを使用すると、ユーザは、プロセスの代わりにオブジェクトに接続できる。好ましい実施の態様においては、該サーバは、オブジェクト管理グループ（OMG）のオブジェクトリクエストブローカ（ORB）プロトコル基準であるCORBAプロトコル、インターネットORB間プロトコル（IIOP）26を提供する。

【0027】

該IIOPを使用すると、該ワークフローシステムオブジェクトを、アプリケーションおよびクライアントツールにさらすことができる。加えて、該IIOPを使用すると、該ワークフローシステムは、現在の技術および将来の技術との統合のためにきわめて拡張可能となる。該Server（サーバ）は、メールを転送し、該ネットワークを管理するソフトウェアを含む。該Mail Adapter（メールアダプタ）28は、多岐に渡るメールシステムに接続してよい。好ましい実施の態様において、該Server（サーバ）は、メールを転送するために、シンプルメール転送プロトコル（SMTP）29またはショートメッセージサービス（SMS）30のどちらかを使用してよい。

【0028】

SMTPは、ホストとユーザセクションの両方との電子メールシステムである。SMTPは、メールを、該メールを分散するメールサーバに送信するための出



eメールプロトコルの遍在する基準である。該SMTPユーザソフトウェアは、多くの場合、パーソナルコンピュータ用のTPC/IPパッケージの中にユーティリティとして含まれている。SMSはテキストメッセージを送受するためのシステムであって、大部分の汎欧州デジタルセルラーシステム(GSM)ネットワークで実現される。SMSは、デジタルページメッセージまたはデジタル電話メッセージを送信するためのプロトコルである。

#### 【0029】

CORBAは、マサチューセッツ州、ファーマンハム(Farmingham, Mass)のオブジェクト管理グループ(OMG)によって定義されている。OMGは、あたかも全体としてここに述べられているかのように、これにより参照して組み込まれているCORBA/IIOP 2.2仕様書を発表した。

#### 【0030】

図2は、骨組およびスタップを作成する際に関係する多様な要素のブロック図を示す。CORBAに対する中心的な概念の一つが、インタフェース定義言語(IDL)の使用である。IDL定義100によって、言語とプラットフォームに依存しないクライアントおよびオブジェクトの作成が可能になる。オブジェクトの骨組110は、IDL定義から作成できる。それから、該骨組は、ある特定のプラットフォームで使用するためのある特定の言語のために開発することができる。該サーバ上の該オブジェクトは、IDLインタフェース170を介して該クライアントと通信する。該骨組がオブジェクトの開発に使用されている限り、該オブジェクトは同じIDL定義を使用して作成される任意のクライアントと通信することができる。

#### 【0031】

クライアントのスタップ120は、IDL定義から作成することが出来る。それから、該クライアントは、ある特定のプラットフォームで使用するためのある特定の言語のために開発することができる。再び、IDLスタップがクライアントの開発に使用されている限り、該クライアントは同じIDL定義を使用して作成される任意のオブジェクトと通信することができる。該サーバとクライアントは、IDLを使用して定義されるインタフェースを介して互いに通信することがで

きる。サードパーティベンダのIDLコンパイラ130は、骨組またはスタップのために、IDLコードをコンパイルされたコードの中にコンパイルする。コンパイルされた骨組コードは、開発時に具体的対象150の中に作りこまれる抽象クラス140を反映する。その場合、該具体的対象が、該ワークフローエンジンの一部となる。コンパイルされたスタップコードは、やはり具体的なクライアント160である抽象クライアントを反映する。該クライアント14は、該サーバ12と通信するために該スタップを使用する。

#### 【0032】

該スタップは抽象および具象の両方である。該開発者は、該スタップを採取し、それをクライアント側で使用し、該クライアント側ではこれ以上のインプリメンテーションはない。サーバ側では、該開発者は抽象を与えられる。それから、該開発者は該骨組を実体化し、具象クラスを作成する。該開発者がスタップを生成しているとき、該開発者は、抽象だけではなく具象も生成している。骨組の場合、該開発者は具象を開発しなければならない。骨組の場合、開発ステップがあるが、スタップの場合には開発ステップはない。骨組の場合、該開発者はインタフェースを定義し、それからそのインタフェースを実際に実現する具象コードを作成する。スタップの場合、該開発者は、該スタップを使用して、該インプリメンテーションに接続する。

#### 【0033】

図1に図示されているように、該クライアントはウェブブラウザ32内で実行する。該ワークフローシステムクライアントは、従来の厚いクライアントアーキテクチャに依存する特別ウェブインタフェースではなく、該ネイティブウェブユーザインタフェースを活用する。該ワークフローシステムは一般的なウェブブラウザを使用するため、特殊ワークフロー関連クライアントソフトウェアのインストールは必要とされない——追加管理費は生じない。好ましい実施の態様においては、該ワークフローシステムは、一般的なウェブブラウザMicrosoft<sup>®</sup> Internet Explorer<sup>™</sup>、およびNetscape<sup>®</sup>のNavigator<sup>™</sup>を活用する。

#### 【0034】

該ワークフローシステムは、ウェブクライアント構造を提供し、それによってユーザは、提供されたワークフローシステムクライアント、または該ワークフローシステムのAPIを使用して開発されたカスタマイズ化されたクライアントを介して該エンジンにアクセスしてよい。アダプタAPIは、ユーザが、該ユーザの構成に特殊なアダプタを開発できるように提供される。その関連するAPIとともにデフォルトのシステムクライアントを提供することは、迅速な顧客対応、拡張および新しいクライアントまたはアプリケーションの合成に対処する。該デフォルトシステムクライアントは、その簡略で学習が容易なユーザインタフェースのために配備、管理およびサポートが容易である。該単純ユーザインタフェースは、自動化プロセスでのユーザの関与を促進する。

【0035】

該ワークフローシステムアーキテクチャは、サードパーティ製品との統合を可能にするためにアダプタをベースにしているため、他のシステムとは異なり、該ワークフローシステムアダプタ技術は、該ワークフローシステムの実質的には任意の既存ITインフラストラクチャとの統合をサポートする。情報システム（IS）インフラストラクチャは、該ワークフローシステムと容易に統合できる。アダプタによって、該Server（サーバ）は、他の構成要素と通信できる。アダプタは、該Server（サーバ）が共通インタフェースに話をすることができるようにする変換器として動作する。言葉の従来の意味においては、まさにアダプタは直接的に接続しない二つのパーツを接続するために使用されるので、これらのアダプタは類似した機能を有する。特定の環境構成に適したアダプタだけをインストールする必要がある。

【0036】

本発明の好ましい実施の態様においては、該Database（データベース）（DB）アダプタは、Javaデータベース接続性（JDBC）基準を使用し、該Server（サーバ）とデータベースサーバの間の通信機構を提供する。（DMS）の目的とは、永続プロセス属性の記憶である。「永続」とは、該ワークフローシステムの電力が停止した場合に、プロセス属性が失われないことを意味する。好ましい実施の態様においては、プロセス属性は、システム停止時にデ

ディスク上に記憶される。該データベース34はプロセス情報を維持する。好ましい実施の態様においては、データベースアダプタは、JDBCを介するMicrosoft SQL Server™およびJDBCを介するOracle®のOracle7™バージョン7.3を含む。

【0037】

該ワークフローシステムが実行を開始すると、三つのオブジェクトが命名サービスから使用できる。すなわち、Document/Directory (文書/ディレクトリ) (DD) フレームワーク20、(DMS) アダプタ22、およびDSアダプタ24である。該DD Adapter (アダプタ) は、該(DMS) アダプタとDSアダプタという二つのアダプタを合わせて一つにしたものである。該DSおよび(DMS) は、自律である。すなわち、互いから別個に存在している。該DDフレームワークの構造は、該(DMS) およびDSがさまざまなベンダ製であることを可能にする。

【0038】

たとえば、該DSを(DMS) から分割すると、Lotus (DMS) というしよにExchange (交換機) ディレクトリを使用できる。(DMS) の目的とは、文書の集中化「返却および貸出し」記憶のためである。該(DMS) は、書式、添付物およびテンプレートを記憶するために使用される。該DDアダプタは、ユーザを認証し、役割分解を実行するために該DSを使用する。好ましい実施の態様においては、サポートされている製品は以下を含む。すなわち、Javaネイティブディレクトリインタフェース(JNDI)を通した軽量ディレクトリアクセスプロトコル(LDAP)、Microsoft Exchange™、およびMicrosoft NT™ユーザレジストリである。

【0039】

該ワークフローシステムは任意の(DMS)と統合できる。(DMS) アダプタは、Windows (登録商標) Microsoft Exchange™ 36、Java開発キット™(JDK) 37、NT™ファイルシステム38のために提供される。DSアダプタは、Microsoft Exchange™ 39、LDAP40、およびWindowsNT™ユーザレジストリ41のため

に提供される。Javaは、プラットフォームに依存しないファイルシステムのビューを提供する。

#### 【0040】

該DDアダプタ、該Directory（ディレクトリ）アダプタ、および該（DMS）アダプタは、それらが可能な限り迅速に立ち上げられるように、その使用を可能な限り単純にするための機能の最小セットを備える。図3は、該DD-（DMS）-DS構造を示す。該DSアダプタは、ディレクトリセッションオブジェクト200を作成する。該（DMS）アダプタは（DMS）セッションオブジェクト210を作成する。

#### 【0041】

該DDアダプタは、つぎに示す機能を備える。すなわち（1）ログイン、（2）ログアウト、（3）該（DMS）オブジェクトの取得、および（4）該ディレクトリオブジェクトの取得である。

#### 【0042】

該Directory（ディレクトリ）アダプタは、以下の機能を備える。すなわち（1）指定されたユーザ向けのディレクトリセッションオブジェクトを作成する、（2）ディレクトリセッションオブジェクトを破壊する、（3）指定されたグループ向けのユーザリストを戻す、（4）一つの長い文字列としての指定されたグループ内のユーザリストを戻す、（5）グループのリストを戻す、（6）指定されたユーザ用のすべてのプロパティ（Name（名前）-Value（値）の組）を戻す、（7）指定されたユーザ用の指定されたプロパティ（Name（名前）-Value（値）の組）を作成する、（8）指定されたユーザ用の指定されたプロパティ（Name（名前）-Value（値）の組）を更新する、（9）指定されたユーザのプロパティの削除である。

#### 【0043】

前記項目（6）、指定されたユーザ用のすべてのプロパティを戻す能力は、該タスクを実行している人物の利益のために、該ワークフローシステムをカスタマイズするために該ワークフローシステムによって必要とされる。前記アイテム（7）および（8）は、それぞれユーザプロファイルを作成し、更新するためであ

る。

【0044】

該(DMS)アダプタは、以下の機能を備える。すなわち(1)指定されたユーザ用の(DMS)セッションオブジェクトを作成する。(2)該(DMS)セッションオブジェクトを破壊する。(3)該指定された経路の文書/ディレクトリの一覧を戻す、すなわちフォルダの文書メンバーを提供する。(4)読取りまたは書き込みのために指定されたファイルを貸し出す。それは、一意のファイル名の付いた該ファイルのコピーを作成し、該構成ファイルに指定されている経路の元で該クライアント機械の上にそのファイルをコピーする。(5)指定ファイルを返す。(6)ファイルが、ある特定のユーザ向けの読取りモードのために貸し出される添付物ディレクトリの経路を戻す。(7)ファイルが、ある特定のユーザ向けの書き込みモードのために貸し出される添付物ディレクトリの経路を戻す。(8)指定された新規ファイルを指定された経路に返す。(9)該D.Dアダプタが実行中である機械の上で利用できるすべてのドライブを戻す。

【0045】

CORBAは、命名サービスの使用に対処する。該命名サービスは、CORBAオブジェクトとクライアント間の層を追加し、その結果該クライアントは、該オブジェクトまたは該オブジェクトが実行中である該サーバのアイデンティティを知る必要がない。むしろ、該命名サービスはこの情報を格納し、該オブジェクトを活用することを求めることのある該クライアントのそれぞれに通知する必要なく、該命名サービスに通知することによってサーバ間でオブジェクトを動き回らせることができる。該命名サービスを該ワークフローシステムで使用することによって、ある特定のインプリメンテーションとは無関係に汎用アーキテクチャが可能になる。該ワークフローシステムは、該命名サービスを活用し、負荷を分散し、分割する。

【0046】

各ワークフローシステム構成要素は、初期化ファイル内にセクションを有する。該初期化ファイルは、各構成要素に、該構成要素が該命名サービスで登録すると、それが、たとえばXなどの該命名サービスによって提供される名前を付け

て登録されることを知らせる。該命名サービスはXを任意の（DMS）オブジェクトに結び付ける。該例で続けると、該初期化ファイルは、該DDフレームワーク構成要素に、該DDフレームワークの名前がZであること、および該DDフレームワークが活用できる構成要素の名前が該（DMS）アダプタに対してはXであり、該DSアダプタに対してはYであることを知らせる。

【0047】

それが、該DDフレームワークが該（DMS）アダプタおよび該DSアダプタについてどのように知るのかである。同様に、該サーバが、それが探している該DDフレームワークを見つけ出すために、該命名サービスは、該サーバが立ち上がるときに、該サーバに、該DDフレームワークがZと呼ばれることを知らせる。その効果は、該ユーザが異なる名前が指定された同じ機械でアダプタの別のセットを立ち上げることができ、それらに互いに依存させることができるという点である。このようにして、該ワークフローシステムは、非常に異なった構成要素による別の構成を有することができる。これにより、複数の構成要素が、同じプラットフォーム上で共存することができる。

【0048】

その結果、該ユーザは複数の構成を有し、ただ該構成ファイルを変更するだけで該（DMS）アダプタを立ち下げ、変更することができる。これがフェイルオーバーに対処する。該構成要素のそれぞれは別々の機械上で実行できる。該構成要素が、ホスト1上のXのように実行中である機械の上で、ホスト1上で何かが破壊されたために誤動作（たとえば、該（DMS）アダプタがダウンした）が発生すると、該ユーザは、ホスト2に該（DMS）アダプタのコピーを転送し、該（DMS）アダプタの名前がXである旨を、ホスト2構成ファイルを介してホスト2に教える。該ワークフローシステムは該命名階層に移動し、旧Xを、結び付けている新Xで置き換え、該システム全体が再び動作する。

【0049】

フェイルオーバーは、通常、災害復旧のためである。該機械がクラッシュし、電源がオフになり、ユーザは別のホストに切り替える必要がある。該特定の構成要素である、たとえば該（DMS）は、実際のオブジェクトリファレンスに結び付

けられている。該DDフレームワークは、該実際のオブジェクトリファレンスに話をする。該DDフレームワークは該名前に関して分解を行う。該構成ファイルは、該DDフレームワークに、該(DMS)アダプタがXと呼ばれることを知らせる。該DDフレームワークが起動すると、それはXを検索し、Xのアドレスを分解し、Xを該実際の(DMS)オブジェクトに関連付け、結び付ける。

【0050】

優位点は、該システムが一度もダウンせずに、週に7日間、日に24時間動作可能であるという点である。フェイルオーバは、連続動作を実行するために必要とされる。このようにしてホスト1が故障すると、ユーザは、ホスト1と同じ(DMS)アダプタを有するホスト2に切り替える。該システムは、該システムが実行中であるプラットフォームの一つが故障したり、破壊されたりするだけで、機能するのを停止しない。このフェイルオーバは、システムアーキテクチャ全体に当てはまる。同様に、該クライアントが該クライアント内で起動すると、該ワークフローシステムは、実質的には該クライアントに対して「これが、あなたが移動し、サーバ上を探している命名サービスである」と伝える。したがって、該ユーザは該クライアントから複数のサーバに選択的にアクセスすることができ、と同時にそれが、該クライアントがどのようにして該サーバにラッチするのかである。

【0051】

したがって、該サーバは機械1でクラッシュしたが、該ワークフローシステムはあらゆる機械名または特定のプラットフォームと完全に無関係である。該ワークフローシステムは、該命名サービスに依存するにすぎない。そして、該命名サービスがクラッシュしたとしても、それは別の機械から立ち上げることができる。ワークフローシステム実施の態様は複数のホストを有し、該Directory(ディレクトリ)アダプタはあるホスト上に置くことができ、該(DMS)アダプタは別のホストの上に置くことができ、しかもサーバがさらに別の機械上に置くことができるため、該柔軟な結び付けは、負荷の分割および釣り合い(負荷の分散)にも対処する。

【0052】



該負荷分割は、該命名サービスの機能ではない。それは、システムがセットアップされる方法、構成部品が使用可能な機械に割り当てられる方法の機能である。該ユーザは、特定なハードウェアプラットフォームとは無関係に任意の命名サービスを活用することができる。「命名サービス」は一般的な用語である。該CORBA命名サービスは、CORBAアプリケーション用である。RMIレジストリは、RMI用の命名サービスである。RMIはJava環境のCORBAの同等物である。

#### 【0053】

図4は、クライアント14と(DMS)要素間の対話を示す。該文書リポジトリ300、(DMS)Impl((DMS)インプリメンテーション)310、および(DMS)Session Object(セッションオブジェクト)320は、別個の(DMS)要素である。ユーザがドロップダウンメニュー330からアイテムを選択すると、アプリケーション論理66がイベントを受け取り、該文書リポジトリに該所望のアイテムを示すメッセージを送信する。それから、該文書リポジトリが(DMS)Implに該選択されたアイテムを実行するように命令する。それから、WF Manager(マネージャ)68およびDS24を活用している(DMS)Implが、該所望アイテムを該サーバ機械上のファイル340に返す(DMS)Session Object(セッションオブジェクト)を作成する。いったんこの動作が行われると、該文書リポジトリは、ファイルを採取し、それを開始するftpユーティリティ350のセットを提供する。

#### 【0054】

図1は、該Script(スクリプト)アダプタ18およびMail(メール)アダプタ28の両方ともが、IIOPインタフェースを介して該ワークフローエンジンに結合されていることを示す。

#### 【0055】

該Script(スクリプト)アダプタ18は、該クライアントで入力されたスクリプトおよび外部サーバ側スクリプトを解釈する。好ましい実施の態様においては、該スクリプトインタプリタは、JavaScriptECMAインタプ

リタ42である。該Mail（メール）アダプタ28は、該サーバとメールサーバ間の通信機構を提供する。好ましい実施の態様においては、SMTPが使用される。該ワークフローシステムの該eメール通知機能が使用されるのであれば、該メールアダプタはインストールされる必要はない。該Notification Facility（通知機構）が、Eメールを通して該ユーザに割り当てられた作業アイテムを通知する。

【0056】

サーバ上に提供されているAPIは、開発者がそれ以外のシステムと動作する専用のアダプタを開発できるようにオープンAPIである。「オープン」とは、ユーザが専用のアダプタを作成できるように、APIが発行されることを意味する。

【0057】

図5は、単一サーバ構成44用の該ワークフローシステムアーキテクチャ10である。好ましい実施の態様においては、該ワークフローエンジンは、スタンドアロンJavaサーバである。該ワークフローシステムサーバは、そのオブジェクトをアプリケーションおよびクライアントツールに露呈するCORBA/IIOPインタフェース26を提供する。該Server（サーバ）は、ステータス変化があった該Client（クライアント）を更新し、要求に従ってテンプレートおよびプロセスデータを更新する。加えて、それは、ユーザにeメール、SMS、および／または事前対応更新を介してタスクを通知する。該ワークフローシステムは、eメールクライアントベースである。

【0058】

該ワークフローシステムサーバは、5個のプロセス、すなわち、ワークフローエンジン、該DBアダプタ、該（DMS）アダプタ、該ディレクトリアダプタ、および該スクリプトインタプリタアダプタから成り立つ。該Server（サーバ）は、プロセス状態データ、プロセスおよび活動関連データ、およびプロセス履歴情報を維持するために該DBアダプタと通信する。該Server（サーバ）は、データベース要求待ち行列を制御する。該Server（サーバ）は、アダプタを介してそれにスクリプトを渡し、結果を受け入れることによって、該S

cript Interpreter (スクリプトインタプリタ) と対話する。

【0059】

該Server (サーバ) は、該ユーザのコンテキストを保持し、(DMS) 形式および添付物が記憶されるロケーションに対するポインタを維持し、認証および役割分解を実行するために該Directory Service (ディレクトリサービス) にアクセスする、該文書/ディレクトリ (DD) アダプタと通信する。ユーザはログインすると、彼らはクライアントidを取得する。該クライアントidを使用して、該ユーザは、それぞれ該ディレクトリアダプタ、または該(DMS) アダプタの中に常駐するディレクトリセッションオブジェクトまたは(DMS) セッションオブジェクトを取得できる。該ディレクトリアダプタおよび該(DMS) アダプタは、それぞれディレクトリセッションオブジェクトおよび(DMS) セッションオブジェクト用の工場である。

【0060】

一方ではアプリケーションと、他方では該(DMS) と(DS) の間のインタフェースは、具象クラスで実現される。該(DMS) クラスおよび該DSクラスは、該クライアント側にインタフェースを有する。該ワークフローシステム(DMS) およびDS APIを使用する優位点とは、ユーザはクライアントアプリケーションを採取し、別の(DMS) およびDSを用いて別のシステム上でそれを実行することができるという点である。該ユーザは、該インタフェースを妨害することなく該アダプタを置換することができる。

【0061】

図5に示されている実施の態様においては、該(DMS) アダプタ22は、該NTファイルシステムと接続するための該NT File (ファイル) システムアダプタである。該ディレクトリアダプタは、該NT Users Registry (ユーザレジストリ) と接続するための該NT User Registry (ユーザレジストリ) アダプタである。該データベース管理システム(DBMS) アダプタ16は、SQLサーバ46と接続するためのシステム照会言語(SQL) アダプタ、すなわちJavaデータベース接続性(JDBC) アダプタである。該スクリプトインタプリタアダプタ18は、該JavaScriptイ

インタプリタ48と接続するための該JavaScriptアダプタである。

【0062】

該ワークフローエンジンは、該DBMSアダプタを通して該DBMSに、ディレクトリアダプタを通して該ディレクトリサービスに、該(DMS)アダプタを通して該文書管理リポジトリに、および該スクリプトアダプタを介して該スクリプトインタプリタにアクセスする。該ワークフローエンジンは、パーシステイティブワークフロープロセスをサポートするために該DBMSを、認証および役割分解を実行するために該ディレクトリサービスを、添付物、書式、およびプロセステンプレートファイルを記憶するために該(DMS)を、サーバ側スクリプトを評価するために該スクリプトインタプリタを活用する。

【0063】

該ワークフローシステムアーキテクチャは、オープンアーキテクチャのある階層システムである。各層で該ワークフローシステムを多岐に渡る要素に開放するために、該Client(クライアント)はウェブブラウザの中で実行し、該クライアントアーキテクチャはJavaアプレットおよびJavaBean構成要素の組み合わせから構成されている。

【0064】

図5は、Internet Explorer 5.2を介したデフォルトのウェブクライアント構成50へのアクセスを示す。該デフォルトのクライアント構成は、モデル層54およびユーザインタフェース(UI)層56という二つの構成要素層を備える。該I I O Pはインタフェースレベルにある。該モデルは、該I I O P層の上に層化されている。該User Interface(ユーザインタフェース)(UI)は、該モデル層上に層化されている。該モデル層は、該Server(サーバ)と該UI間の中間層として見なすことができる。該モデル層は、該クライアントオブジェクトの状態をカプセル化し、CORBA I I O Pを介して該Server(サーバ)と対話する。該UI構成要素は、ウェブクライアントプレゼンテーション用であり、該ユーザの視覚的なニーズを満たすためにカスタマイズすることができる。

【0065】

ユーザがすでにUIを有している場合には、該ユーザは、該モデル層の中に該ワークシステムAPIを必要とするだけである。該ウェブクライアント (Model (モデル) およびUI構成要素) は、Sun Microsystems<sup>®</sup> のJava Studio<sup>™</sup>などのサードパーティウェブ開発ツールのパレットの中で容易に構成し直され、拡張またはカスタマイズすることができる。ウェブクライアントは、クライアント構成要素のモジュラリティおよびカスタマイズアビリティによって容易に認識できる。該クライアント構成要素のモジュラリティおよび機能内訳が、容易な再構成に対処する。該Bean構成部品は、製造者ツールを通したカスタム化および拡張に対処する。

#### 【0066】

図5では、該クライアントは、HTTPプロトコル60を介してインターネット情報サーバ58と通信する。該HTTPインタフェースとは、動的に構成されているウェブページをクライアントに、およびクライアントから提供するのと担当するURLベースのインタフェースである。インターネット情報サーバのアクティブサーバページ (ASP) 62は、CORBA IIOPを介して該ワークフローシステムサーバと通信する。

#### 【0067】

図6は、ウェブクライアントのデフォルト構造を示す。該ウェブクライアントは、Java アプレットおよび/またはJava Bean構成要素の組み合わせを備える。これらのクライアント構成要素は、JavaスクリプトおよびHTML 64を使用していっしょに配線される。該クライアントは、すべてのジャバビーンズがスタンドアロンで動作できるようにコンポーネント化されている。インプリメンテーションソリューションは、ジャバビーンズのすべてまたはいくつかを使用することができる。

#### 【0068】

該ワークフローシステム以外のシステムはウェブクライアントを有するが、それらは真にオープンではない。それ以外のシステムのクライアントは、独占的なシステムに結び付けられている制限を有し、大部分は移植の問題のために制限される。それ以外のワークフローシステムは、それらがソフトウェアをユーザ機械

上にインストールすることを必要としたために成功しなかった。それらは、ウェブベースである代わりにクライアントサーバベースであった。該ワークフローシステムは、ユーザ機械上よりむしろウェブサーバ上にそのクライアントを有する。

【0069】

JavaBeansによって、ユーザは、ソフトウェアユーザインタフェース構成要素を柔軟にアセンブルし直すことができ、それによってユーザはユーザのニーズに最もよく適するワークフローシステムを作成することができるため、該ワークフローシステムClient（クライアント）の構成要素アーキテクチャは、JavaBeansに基づいている。該ビーンズによって、該ユーザは、その管理を含む該ワークフローシステムのすべての面にアクセスし、操作することができる。該ワークフローシステム機能性は拡張することができる、および/またはサードパーティ製品およびツールと統合することができる。JavaBeansはプラットフォームから中立であるため、開発者は任意のアプリケーション環境内で任意のOSのために開発することができる。

【0070】

該クライアント構成要素は、厳密には二つの層、すなわちモデル構成要素層とUI構成要素層の中で構造化されている。モデル構成要素は、CORBA IIOPを介して、該クライアントオブジェクトの状態およびサーバとの対話をカプセル化するために使用される。該クライアントUI構成要素の目的とは、オーガナイズ、プロセスのサムネイルオーバービュー、または詳細なビューなどの該ワークフローシステムユーザインタフェースのさまざまな構成要素を提供することである。これらの構成要素は、クライアントオブジェクトを維持しない、あるいはワークフローシステムサーバとの対話を実行しない。

【0071】

該クライアント構成要素は、NetscapeからVisual JavaScript、あるいはSun MicrosystemsからVisual Studioなどの視覚的な開発ツールのパレットに容易にインポートできるように設計されている。このような構成要素をサードパーティツールのために使用でき

るようにすることによって、該クライアントの容易な再構成、拡張およびカスタマイズが可能になる。このようにして、ユーザがまったく新しいクライアントを最初から作成することを望まない場合には、ユーザはデフォルトのワークフローシステムをカスタマイズし、それによってカスタムクライアントを作成することができる。

#### 【0072】

該モデル構成要素は、アプリケーション論理 (AppLogic) 66、ワークフロー (WF) マネージャモデル68、形式モデル70、作業アイテムモデル71、プロセスモデル72およびテンプレートモデル74を含む。図6では、HTMLおよびJavaScript以外の括弧の間の名前がクラス名である。該Application (アプリケーション) 論理モデルは、該クライアントの全構成要素間の調整およびイベントのディスパッチを担当する。該WFManagerモデルは、ログイン/ログアウトおよび該サーバからのリストの検索を担当する。該Form Model (書式モデル) は、形式とプロセス構成要素間のデータ交換を担当する。該Workitem Model (作業アイテムモデル) は、作業アイテム、すなわち処理されるのを待機する任意の活動を担当する。該Process Model (プロセスモデル) は、プロセスインスタンスを担当する。該Template Model (テンプレートモデル) は、計画、すなわちプロセステンプレートを担当する。

#### 【0073】

包括的なレポート作成をサポートするために、該UIは、該Menu Bar (メニューバー) およびTool Bar (ツールバー) 76、該Organizer (オーガナイザ) 78、該Overview (オーバビュー) 80、該Process (プロセス) /テンプレートビュー/エディタ82、該Activity (活動) 制御バー84、該Activity (活動) 詳細ビュー85、該Activity (活動) 応答バー86および形式ビュー88から構成されている。

#### 【0074】

該Organizer (オーガナイザ) は、プロセス、プロセス定義および作

業アイテムのリストを提示する。該Overview（オーバビュー）は、選択されたプロセスまたはプロセス定義の縮小サムネイルオーバビューを提示する。該サムネイルオーバビューは、選択されたプロセスまたはプロセス定義のグラフィック表記である。該Process（プロセス）／テンプレートビュー／エディタは、プロセスおよびプロセス定義の表示／編集を可能にする。該活動バーは、添付物および形式に対するアクセスを可能にする。該応答バーは、該ユーザが特定の選択肢を用いて活動の完了を指定できるようにする。該形式ビューは、ユーザ定義書式の内容を提示する。

#### 【0075】

該Thin Client（薄いクライアント）とは、該Full Client（完全クライアント）のサブセットである。該活動詳細ビューは、該活動バー84、該応答バー86、および該形式ビュー88を備える。本発明の実施の態様は、表示されている薄いクライアントビューまたはグラフィックビューのどちらかを有してよく、そこではそれらは、同時に共存しない。しかしながら、該ユーザは、該同じ活動に関して、該薄いクライアントビューと該グラフィックビューの間で切り替えることができる。該薄いクライアントビューおよび該グラフィックビューは、それらが活動を表示する、異なる方法であるため、互いに置換することができる。該3個の該薄いクライアントまたは該グラフィックビューが、該Developer Client（開発者クライアント）画面レイアウトのターゲット領域を備える。本発明の別の実施の態様は、同時に表示されている該薄いクライアントビューと該グラフィックビューの両方を持つてよい。

#### 【0076】

該ワークフローシステムにより、知識作業者は、スクリプトまたはプログラミングを実行しなくてもプロセスモデルを作成することによって有意義なワークフロープロセスを定義できる。これらのプロセスモデルは、機能が豊富なグラフィックユーザインタフェースを用いて、プロセス起動前、または「実行中」のどちらかに、容易に視覚的に作成、修正される。プロセスモデルは、プロセスインスタンスを介して実現される。プロセスインスタンスは、それによってワークフロープロセスがモデル化される協力媒体である。プロセスインスタンスは、計画テ



ンプレートから作成される。プロセスインスタンスは、プロセス属性と呼ばれている共用データ空間および（プロセス定義またはテンプレートとしても知られる）計画から構成されている。プロセス属性は、永続データ要素または文書などの外部情報ソースへのリンクである場合がある。プロセス属性は、集散的にプロセス関連データと呼ばれることもある。

#### 【0077】

プロセスと活動の両方ともアクティブであってよい。アクティブプロセスまたは活動とは、人間の応答を待っているものである。多くのプロセスインスタンスが、任意の一時点でアクティブとなり、同じまたは別の計画テンプレートから作成することができる。プロセステンプレートは、オフラインで編集することができる、オンラインで動的に編集することができる。プロセス変更制御は、動的にプロセスインスタンスおよび計画テンプレートを更新する。該プロセスインスタンスは、そこからそれらが作成され、それによって、ユーザが、プロセスインスタンスの特徴を修正することにより実行時に計画テンプレートを修正できる計画テンプレートに関係し続ける。

#### 【0078】

プロセスインスタンスは、パーソナルテンプレートを作成するために使用できる。実現されたワークフローアーキテクチャ内の該CORBAインタフェースおよびJAVA言語の該プラットフォームに依存しない性質は、プロセスインスタンスが、多くの異なるコンピュータプラットフォーム上にあることを可能にし、プロセスインスタンスと計画テンプレートの両方の動的な修正を可能にする。該ワークフローシステムは任意の一つのコンピュータシステムに結び付けられていないため、該ワークフローシステムは、テンプレートおよびプロセスのサイズに何のサイズ制限を課さない。

#### 【0079】

計画は、矢印を介してネットワーク化されているノードのセットから構成されている。各ノードは活動（ビジネス活動）またはフロー制御演算子を表す。矢印は二つのノード間の経路を表す。活動は、その出矢印上で、それらがアクティブ状態（実行される準備が完了している）にあるときにイベントを作成することが

でき、複数の活動が一度にアクティブであってよい。異なるノードタイプは、イベントに対し応答する上でさまざまな動作を有してよい。ノードには、それがユーザの関与を必要とする活動を表す場合に役割を割り当てることができる。役割はユーザと関連しており、組織ディレクトリ内で維持される。複数のユーザが同じ役割を割り当てられてよく、ユーザは複数の役割を果たす。開始ノードと呼ばれている計画の開始点を特定する特殊ノード、および終了ノードと呼ばれている計画中の終了点を特定する別のノードがある。有効な計画は一つの開始ノードを有するだけでなければならない。計画には複数の終了ノードがあってもよい。ノードはそれ自体の属性を有する。

#### 【0080】

その計画を含むプロセスのすべての面は、そのライフサイクルの任意の段階で修正されてよい。プロセスは不完全な計画を使用して作成できる。プロセス計画を任意のときに修正、拡張することは、該ワークフローシステムの顕著な機能の一つである。プロセスインスタンスが作成されると、それには実施する計画が割り当てられる。プロセスインスタンスは、実行中の計画と考えることができる。計画は静的な再利用可能なプロセスモデルである。プロセスインスタンスは作成された後、それは手動でまたは自動的に起動できる。プロセスの起動により、その開始ノードはアクティブになり、その出矢印上でただちにイベントを生成する。異なるノードタイプは、その入矢印の内の一つに到着するイベントに対応する際にさまざまな動作を有してよい。

#### 【0081】

該ワークフローシステムは、Start Node (開始ノード)、Exit Node (終了ノード)、Condition Node (条件ノード)、OR Node (論理和ノード)、AND Node (論理積ノード)、Activity Node (活動ノード)、Subplan (サブプラン)、およびSplit Activity Node (分割活動ノード)を含む所定ノードタイプのセットを有する。各ノードには専用の属性がある。該Start Node (開始ノード)は、計画の開始点をはっきり分ける。任意の計画には開始ノードがなければならない。該Exit Node (終了ノード)が計画の終了点を

はっきりと分ける。該終了ノードによって、該プロセスは、それがその入矢印のうちの一つでイベントを受け取ると終了する。該Condition Node (条件ノード) は、条件分岐を提供する。該Condition (条件) ノードは、指定された基準が満たされない限りプロセスフローを停止するプロセスフロー制御ノードである。

#### 【0082】

その出矢印のそれぞれ一つを、条件と関係つけることができる。該条件は、指定されたプロセス属性を基準とした相対表記として指定される。条件ノードがその入矢印の一つでイベントを受け取ると、それは、その相対表記が真に対し評価される第1出矢印でイベントを生成する。すべての表記が真に対し評価できない場合、イベントはデフォルト矢印に沿って発せられる。該OR (論理和) ノードは、到達するとすぐにプロセスフローを継続するプロセスフロー制御ノードである。該OR (論理和) ノードは、それがその入矢印のどれかでイベントを受け取ると、すべての出矢印でイベントを生成することによって並行分岐を開始する。該AND (論理積) ノードは、強制的に活動を動機させるプロセスフロー制御ノードである。該AND (論理積) ノードは、それがその入矢印のそれぞれからイベントを受け取るとき、および受け取るときだけ、すべてのその出矢印でイベントを生成することによって並行分岐に加わる。

#### 【0083】

該Activity (活動) ノードは、ユーザ対話を必要とする活動の図の表記である。この種のノードがアクティブになると、該ノードに割り当てられた役割のすべてのメンバーが通知される。通知は、(該ユーザのプロファイルに従って) クライアントへ、またはEメールを介して送ることができる。該活動ノードには、該含有プロセスの属性へのユーザアクセスを可能にするために一つまたは複数の形式が割り当てられる。活動には、添付物として一つまたは複数のファイルも割り当てられてよい。該Subplan (サブプラン) ノードによって、該ユーザは既存の計画を別の計画の本体の中に再利用することができる。該Split Activity Node (分割活動ノード) とは、実行時に自動的に複製される活動ノードである。活動複製のそれぞれが、該ノードに割り当てら

れている役割のメンバーの一つに割り当てられる。

【0084】

事前に定められた矢印タイプの該Workflow (ワークフロー) システムセットは、その前任者ノードによって生成されるイベントをその後継者ノードに伝搬することにより、二つのノードを繋ぐ矢印を含む。各矢は、ラベルプロパティを有する。

【0085】

各ノードには専用の属性がある。すべてのNode (ノード) タイプには、以下に示す事前に定められたプロパティがある。すなわち、Label (ラベル)、PrologスクリプトおよびEpilogスクリプトである。各ノードには、プロセスのように、ユーザ定義の属性がある場合がある。ユーザ定義属性は、ユーザが設計時に指定する名前および値タイプを指定できる。該Activity Node (活動ノード) およびSplit Activity Node (分割活動ノード) には、以下の追加プロパティがある。すなわち、Role (役割)、Description (記述)、Forms (書式)、Attachments (添付物)、およびPriority (優先順位) である。あらゆる活動ノードには、選択肢と呼ばれている動的プロパティがある。該選択肢は、該ノードの出矢印を列挙する。該選択肢プロパティは、ユーザが、イベントが伝搬される矢印に移動できるようにするために使用される。

【0086】

該ワークフローシステム内では、タスクは、「役割」を通してユーザに割り当てられる(役割とは、たとえばLDAPなどの組織ディレクトリ内に含有されるユーザグループなどのユーザの集合の論理的なグループ分けである)。該ワークフローシステムは、Dynamic Role Assignment/Management (動的役割割当て/管理) によるタスク割当てでの変更または不確実性を容易に処理することができる。すなわち、ユーザのタスクへの該動的割当ては、位置、作業負荷、またはスクリプト機能を通したそれ以外の何らかのユーザ定義変数に依存する。万一あるタスクに割り当てられている人物が利用できない場合には、この機能は、プロセスが遮られずに続行できるようにする。それ

は、人員の変更があるたびにプロセス定義を更新するというニーズも排除する。プロセス定義中、および開始後に人員を割り当てるまたは割り当てし直す柔軟性は、ワークフローソリューションの維持を大幅に簡略化する。

#### 【0087】

設計時の活動が役割に割り当てられる。実行時、その役割はユーザに分解される。該ワークフローシステムは、誰がその役割を実行できるのかを確かめるために該ディレクトリを検索する。該役割の目的とは、該活動ノードによって表される該タスクを実行することができる、および／または実行するように許可されているユーザを特定することである。これらのユーザは、活動のAssignees（譲受人）とも呼ばれる。各活動ノードまたは分割活動ノードに役割が割り当てられる。譲受人は、活動に取り組むためにその割当てを通知される。

#### 【0088】

任意のワークフローシステムユーザは、複数の役割に割り当てることができる。該役割情報は、該ユーザディレクトリサービスで維持される。該ワークフローシステムサーバは、ユーザを確認し、彼らの割り当てられた役割およびプロセス、活動およびその属性にアクセスし、操作する上での彼らの能力を特定するために、該ディレクトリサービスにアクセスする。役割の使用は、人員の変更があるときにあらゆるプロセス定義を修正するニーズを排除する。代わりに、該Directory Service（ディレクトリサービス）内でのグループ定義だけを更新する必要がある。

#### 【0089】

活動ノードは、事務員またはmanager Ofなどの文脈依存役割などの静的な役割が割り当てられてよい。文脈依存役割は、JavaScriptとして指定される。該ジャバスクリプトは、ノード起動時に評価される。該スクリプトの評価は、該活動に割り当てられる一人または複数人のユーザを戻す。該ワークフローシステムは、文脈依存役割分解のインプリメンテーションをサポートするJavaScriptコマンドのセットを提供する。

#### 【0090】

サブプロセスとは、具体化され、アクティブである活動ノードと動的に関連付

けられている計画のことである。サブプロセスが活動ノードと関連付けられると、該活動ノードは一時停止され、該関連付けられたサブプロセスが、親活動によって表されるタスクを実行するために起動される。この機能は、該活動の目標を達成するために最良の行動方針を決定するために活動を実行することを担当するユーザに自主性を与える。プロセスプランが複数の組織に浸透するとき、すべての組織がある特定の活動を実行する上で同じように動作すると仮定することは実際的ではない。

#### 【0091】

スクリプトは、その機能性を拡張し、ユーザ対話を必要としない活動の自動化を提供するために、計画およびそのノードと関連付けることができる。スクリプトは、サーバによって評価され、好ましい実施の態様においては、JavaScriptスクリプト機能言語内にある。スクリプト機能は、タスクを自動化し、それによってタスクをユーザの関与なしに実行できるようにする。スクリプトは、プロセスの属性を操作するために作成することができ、スクリプトは、外部アプリケーション/システムへのアクセスを可能にし、それによってユーザ定義拡張フレームワークを提供する。

#### 【0092】

スクリプト機能とは、グラフィックデザイナーを補完するために必要とされる言語である。グラフィックパラダイムにより、ユーザは、本質的にアルゴリズム的、手続き的あるいは算術的である何かを容易にかつ実践的に定義することはできないことがある。このようにして、ユーザは、多くの場合「 $x + y$ 」などの算術式を図式によって設計するのに苦勞する。しかしながら、スクリプト機能言語では、算術式を行うことははるかに容易である。スクリプト機能によって、標準スクリプト拡張を介したプロセス属性へのアクセスが可能になる。ユーザは多くの場合ある特定のプロセスの属性に対して算術演算を実行することを希望するため、該提供されたスクリプト拡張により、ユーザはプロセス属性にアクセスする。スクリプト機能は、該ワークフローシステムフレームワークを通してユーザが外部システムに到達できるための別の手段でもある。

#### 【0093】

それに対するパブリックメソッドを有するジャバクラスは、異なるアプリケーションに到達できるために、JDBC、IIOPまたはDCOMを使用して設計できる。DCOMはMicrosoftの分散COMオブジェクトである。COMは構成要素の略である。Microsoftは、本質的にはJavaに非常に類似しているが、それは固有である構成要素モデルを有する。DCOMは、分散形構成要素モデルであり、遠隔アイテムにアクセスするための方法である。DCOMおよびIIOPは、ユーザに、さまざまなアプリケーションへのアクセスを与える。アプリケーションがDCOMに結び付いている場合、ユーザはDCOMにアクセスすることができる。

【0094】

考え方は、ユーザが、それがデータベースにアクセスするためにJDBCを活用しようと、IIOPアクセス可能アプリケーションにアクセスするためにIIOPを活用しようと、あるいはDCOMアプリケーションにアクセスするためにDCOMを活用しようと関係なく、Javaクラスを構築するということである。該ユーザは、該サーバに該ジャバクラスを該スクリプトアダプタにロードするように命令するために構成ファイルを入力する。

【0095】

該(DMS)およびDSは、ユーザが特にこれらのためにアダプタを必要とするほど重要である。該スクリプトアダプタは、それ以外の種類のアイテムにも必要とされる。該Script (スクリプト) アダプタは、文脈依存役割分解で重要な役割を果たす。

【0096】

プロセスを設計する際に、ユーザはオーナー役割および／またはオーナー役割スクリプトを指定してよい。ある特定の活動の場合、該プロセスオーナーは、譲受人役割および／または譲受人役割スクリプトを指定してよい。該プロセスオーナーは、該譲受人役割と譲受人役割スクリプトの両方を使用することは要求されていないが、それらの任意の組み合わせを使用してよい。活動に譲受人役割がなく、譲受人役割スクリプトもないとき、該役割は該プロセスの該オーナーに割り当てられる。活動に譲受人役割 (たとえば、Role (役割) : クレーム調整者) があるが

、譲受人役割スクリプトがないときには、該プロセスオーナーは、自分が、誰が該活動に取り組む必要がある人々であるのかを静的に知っており、彼らの内の一人が、たとえば該クレーム調整者のメンバーである人物に活動に割り当てられることを要求するなど該活動を実行しなければならないと言っている。

【0097】

該クレーム調整者の役割は、任意の特定のクレーム調整者が活動进行处理することが必要ではないためこのケースでは十分である。対照的には、プロセスの後半のポイントで、該ユーザが、活動を行った該人物A1が、活動を行わなければならない人物、A2でもあると言いたいと仮定する。このような場合は、スクリプトを必要とする。それは、スクリプトが活動を行ったA1を見つけ出し、活動する人A2を割り当てようとする場合である。活動A1を行った同じ行為者が、活動A2を割り当てられる。

【0098】

該行為者とは、実際にタスクを実行した人物である。譲受人が行わなければならない活動A1を行ったことを除いて、特定な資格はないため譲受人役割は指定されていない。しかしながら、譲受人役割スクリプトは、活動A1を実行する人物が活動A2を実行することを確実にするために指定されなければならない。該譲受人役割/譲受人役割スクリプトの組み合わせの最後の順列の中では、該譲受人役割と譲受人役割スクリプトの両方とも指定されている。このケースでは、組織内の誰もが承認を求めることができる。該スクリプトは、該行為者をFIG. outするために実現され、承認は該行為者の該マネージャによってであるが、該プロセスの該オーナーは、プログラミング間違いがないこと、該スクリプトにより指定される人物が一定の資格を有し、それが、該役割が入ってくる場所であることを確認することを希望する。

【0099】

該役割は承認許可、すなわち署名の権限を有する。この役割のメンバーであるものは誰でも署名の権限を有する。該譲受人の最終的な決定は、該スクリプトを満たす者と、該役割を満たす者の共通部分である。役割分解は、役割ごとに、あるいは役割スクリプトごとに発生する。同様に、誰がプロセスのオーナーとして指



定されようとも、該オーナの役割および／またはオーナ役割スクリプトを満たさなければならない。

#### 【0100】

図7は、該ワークフローシステムスクリプトフレームワークのブロック図を示す。該スクリプトアダプタ18は、該ワークフローエンジンにスクリプト言語インタプリタ42を接続するためのアダプタである。好ましい実施の態様においては、該スクリプトインタプリタは、サーバ側スクリプトインタプリタである。該スクリプトインタプリタは、スクリプトコマンドを処理する。該スクリプトインタプリタの目的とは、Dynamic Invocation Interface（動的呼出しインタフェース）（DII）／IIOPまたはJava拡張子を介して外部アプリケーションへのアクセスを提供することである。該スクリプトインタプリタは、クライアントおよび外部サーバ側スクリプトを通して入力されたスクリプトを解釈することである。

#### 【0101】

該スクリプトインタプリタは、外部サーバ側CORBAオブジェクトの呼出しを可能にする。該事前に定義されたスクリプトインタプリタコマンド400は、スクリプトを書き込むことを希望するユーザに、プロセス、活動またはシステムの属性へのアクセスを提供する。該ユーザ拡張子フレームワーク410は、カスタマが新規スクリプトコマンドをJavaスクリプトインタプリタに追加できるようにし、カスタマがそのメソッドがJavaスクリプトインタプリタ内からのコマンドとして使用できるJavaクラスを開発できるようにする。該ユーザ拡張子フレームワークの目的とは、カスタマが外部データベースおよびアプリケーションにアクセスできるようにし、このようにしてプロセスのユーザ定義属性と外界の間のデータフローを可能にすることである。

#### 【0102】

好ましい実施の態様においては、Javaネイティブメソッド呼出し（JNI）が、Javaスクリプトインタプリタでユーザ拡張子フレームワークを実現するために使用される。JNIは、Javaアプリケーションと非Javaアプリケーション間の相互運用性を可能にするためにSunによって提供される技術で

ある。好ましい実施の態様においては、該スクリプトインタプリタは、JavaScript ECMA言語を使用する。該Virtual Machine（仮想機械）420は、Java言語インタプリタである。Javaは、コンパイルされた言語であるC++に対照的な解釈された言語であり、Javaは解釈された言語であるため、Javaコードは移植可能である。

#### 【0103】

クライアント側スクリプトは、ワークフローシステム形式に使用される。クライアント側スクリプト機能は、ブラウザに依存する。どのようなスクリプト機能がブラウザから使用できようとも、該ワークフローシステムから利用できる。クライアント側スクリプトは、ブラウザによってサポートされるあらゆるスクリプトであってよい。該ワークフローシステムは、該ワークフローシステム形式と関連付けられているスクリプトが、プロセスのおよび活動の属性にアクセスできるようにする追加スクリプト構成要素を提供する。

#### 【0104】

プラットフォームに依存する機能性を備えたクライアント側スクリプトの拡張は、ブラウザから利用できる。それが該ブラウザにとってネイティブであるのか、プラグ接続を必要とする何かであるのかに関係なく該ブラウザから利用できる該スクリプト機能は、該ワークフローシステムを通して利用できるようになる。ユーザがクライアント側で使用する事が許されているスクリプトとは、該ブラウザが提供するものすべてまたはユーザがそれを用いて該ブラウザを拡張できるものすべてである。

#### 【0105】

好ましい実施の態様においては、該ワークフローシステムは、ヨーロッパコンピュータ製造業者協会（ECMA）規格のJavaScriptを使用する。該ECMA規格内蔵コマンドに加えて、該ワークフローシステムは、プロセスおよび活動の多様な面にアクセスし、操作するための追加コマンドを提供する。該ワークフローシステムは、サードパーティインテグレータが独自の拡張子でJavaScriptコマンドを拡張できるようにする。JavaScriptコマンドのセットは、二つの方法で拡張できる。

## 【0106】

すなわち(1)利用可能な関数に基づいて新しいJavaScript関数を作成し、ファイルに保存してから、このような関数を、それをJavaScriptインタプリタにロードすることにより利用できるようにする、および(2)そのメソッドをサーバ側JavaScript内から呼び出すことができるジャバクラスを作成する。該ジャバクラスメソッドは、外部ソースまたは外部アプリケーションへのアクセスが必要とされる場合に使用される。これらの拡張子の目的とは、該ワークフローシステムスクリプトが、受け継いだアプリケーションおよび外部データ格納にCORBAおよびJavaを通してアクセスできるようにすることである。

## 【0107】

スクリプト機能の利点は、ジャバクラスの作成、ジャバクラスのロード、およびスクリプトがジャバクラスへアクセスできるようにすることを含む。該JavaScript Interpreter (JavaScriptインタプリタ)は、Javaクラスを該スクリプトアダプタの中にロードすること、およびそれらにスクリプトがアクセス可能であることを可能にするために内蔵関数を含む。プロセス関連スクリプトである2種類のスクリプトは、Owner (オーナー) スクリプトおよびRole (役割) (譲受人) スクリプトである。該Owner (オーナー) スクリプトは、プロセスインスタンスのオーナー (複数の場合がある) を計算する。Assignee (譲受人) スクリプトとしても知られている該Role (役割) スクリプトは、活動の譲受人 (複数の場合がある) を計算する。これらのスクリプトの両方とも、該ワークフローシステム内での動的役割分解を可能にする。

## 【0108】

2種類のスクリプトが計画に指定されてよい。これらの計画レベルのスクリプトは、InitスクリプトおよびCommit (コミット) スクリプトである。該Initスクリプトの目的とは、プロセス属性の初期化である。該Initスクリプトは、プロセスインスタンスが作成されると一度評価される。該Commit (コミット) スクリプトの目的とは、任意のトランザクションの開始前にプ

プロセス属性を妥当性検査することである。該InitスクリプトおよびCommit (コミット) スクリプトは、両方ともサーバ側スクリプトである。

#### 【0109】

2種類のスクリプトが、活動および条件ノード、すなわちPrologスクリプト (Body (ボディ) スクリプトとも呼ばれる) およびEpilogスクリプトに指定されてよい。該Prologスクリプトはノード起動時に評価されるが、Epilogスクリプトはノード完了時に評価される。エピログスクリプトはOR (論理和) ノードおよびAnd (論理積) ノードにも指定されてよい。該PrologスクリプトおよびEpilogスクリプトは、サーバ側スクリプトである。

#### 【0110】

書式作成は単純であり、データベースの知識は必要としない。過去においては、ユーザがプロセス属性を取り扱うために知らなければならなかった多数の専門用語があった。過去においては、プロセス属性を定義する前にとらなくてはならない、いくつかのステップがあった。データベースのデータは定義されなければならなかった。テーブルフィールドは指定されなければならなかった。およびテーブルは初期化を必要とした。テーブル内の行は、テーブルがプロセス内から使用できるようになる前に初期化されなければならなかった。

#### 【0111】

好ましい実施の態様においては、プロセス属性は図表で定義され、データベース専門用語とは無関係である。該ワークフローシステムは、ユーザがデータベース専門用語を知っていることを要求しない。データベースは多くの特異な表現法を有しているが、該ワークフローシステムはプロセス属性を図表で定義するため、プロセス属性定義は、データベースとはまったく無関係である。

#### 【0112】

HTML/JavaScript書式または任意のそれ以外のウェブアプリケーションまたは文書への経路は、活動ノードと関連付けることができる。書式は、任意のワークフロー管理システムで不可欠であり、該ワークフローシステムとその他のアプリケーションの間で情報を交換する手段となる。書式は、該書式

上に表示されるプロセスデータアイテムを指定することによって作成される。それらは、プロセス定義またはインスタンスと関連付けることができる。好ましい実施の態様においては、該ワークフローシステムはHTML/J a v a S c r i p t 書式ルーティングを使用する。他方、添付物は、完全に静的な情報を提示するためだけに役立つ自由形式の事前に存在している文書である。添付物は、プロセスインスタンスと関連付けられるだけである。

【0113】

Form Editor (書式エディタ) は、フロー設計者が、ワークフローシステムプロセス設計ツールから自動的に生成される書式をカスタマイズし、拡張できるようにする複数のサードパーティツールと統合される。

【0114】

該ワークフローシステムプロセス設計ツールは、プロセスのグラフィックデザイナーである。「プロセス設計者」および「グラフィックデザイナー」は、交換可能な用語である。該プロセス設計者は、ユーザが形式の要素を指定するのを助ける。

【0115】

FormBuilder (フォームビルダ) には二つの要素がある。すなわち (1) 書式内にあるプロセス属性とは何か、データ内容である。および (2) プレゼンテーションとレイアウトである。該書式モジュラリティは、柔軟に置換可能な構成要素に対処する。該グラフィックデザイナーツールは、ユーザが、該ユーザによって指定される内容に基づき書式を作成するのを助ける。該ユーザは、どのプロセス属性をユーザが書式上に所望するのかを指定する。それから、該グラフィックデザイナーは、HTML 書式を生成し/ユーザに与える。該ワークフローシステムグラフィックデザイナーはHTML 書式を製作するため、該ユーザは、HTML 書式をカスタマイズするために、多数の市販されているツールを使用することができる。

【0116】

該グラフィックデザイナーは、該ユーザが要素を検索し、それらを更新できるように、指定されたデータ要素がその上にあり、SDK実行ファイルへのフック

のあるHTML書式を生成する。該書式は、SDKを通して、ユーザ指定要素およびサーバから自動的に生成される。ウェブオーサリングツール向けの大きな業界がある。AdobeのPage Mill、Front Page、Net Fusion、およびVisual Pageは、すべて、html書式を採用し、ユーザに、該ユーザが好むレイアウトと表記を設計させることのできるウェブオーサリングツールである。これは、書式をよく見せる一つの方法である。ユーザは、ツールを使用しなくても、該書式を採取し、JavaScriptおよびHTMLをそれに適用することができる。HTML書式の該自動生成は、該ワークフローシステム向けの値チェーンを作成する。

【0117】

該書式は、該プロセスの属性を拡張できるようにユーザが該サーバと接続できるようにするだけでなく、該書式は外界とのデータを交換するための手段でもある。これらのツールを通して、またはJavaおよびhtmlスクリプト機能を通して、該ユーザは、該サーバを通して該プロセスに一種結線／接続される一団のフィールドで生成された生成済み形式を取り、ウェブオーサリングツールを使用するか、ユーザが外部データソースと対話するまたは外部データソースとデータを交換できるようにするスクリプト機能を使用するかのどちらかで新規データ要素を追加することができる。この自動書式生成は、独自の書式ビルダツールに対するニーズを排除した。このようにして、これらの書式をカスタマイズ、開発および拡張する開発者は、ウェブオーサリング向けの標準的な種類の知識であるhtml業界全体での実験を行えばいいだけである。

【0118】

書式生成は自動的である。革新が、該ユーザが市販されている技術を利用できるように、html書式の自動生成を行っている。該ユーザは、ウェブオーサリングツールを使用して該書式にいくつかの要素を置いたり、スクリプト機能を使用して外界とデータを交換することができる。データは、別のデータベースと、またはPeopleSoft等の製品と交換されてよい。スクリプト機能またはツールを通して、該ユーザは、外界とプロセス間でデータを交換することもできる。

## 【0119】

まさにHTML書式の生成がその他の技術を利用するように、該クライアントの該構造は、クライアントの大半がそのコアであるHTMLにあるためにそれ以外の技術を利用する。たとえば、該薄いクライアントは、基本的にはいくつかのアプレットを備えたhtml形式である。その結果、該薄いクライアントは、オーサリングツールの中に取り込み、カスタマイズすることもできる。

## 【0120】

ユーザは、そのアプリケーションに合わせるために書式をカスタマイズすることができる。該ワークフローシステムは、単純で簡素な書式を生成するための機構を提供する。該書式を装飾するため、および他のアプリケーションへのフックを作成するために、ワークフローによって生成された書式を、任意のHTML書式エディタなどのHTMLページを修正できる任意のツールの中に引き入れることができ、所望の変更を加えることができる。

## 【0121】

書式をカスタマイズするには制限がある。実質的には、該ワークフローシステムにより作成される該html書式を修正するには、任意のウェブオーサリングツールまたはテキストエディタさえも使用することができる。しかしながら、唯一の要件とは、最初に書式が該ワークフローシステムを用いて作成され、該書式内のフィールドの意味も、該書式のファイル名とロケーションの改変されないということである。該書式は、以下の二つの理由から該ワークシステムを用いて作成されなければならない。すなわち(1)該ワークシステムは、該テンプレートのユーザ定義属性の適切な意味を捕捉し、これを該書式の中に埋め込む、および(2)該ワークフローシステムは、該書式を修正するためにどの手段が使用されたのかに関係なく、活動がアクティブになると、このコンテキストが思い出され、該書式を期待されるように、該書式のコンテキストを思い出す。

## 【0122】

該ワークフローシステムにより生成される該書式は、該ワークフローServer(サーバ)機械に記憶され、フィールドの単一列から成り立つ。これらのフィールドは、ユーザ定義テンプレート属性(データアイテムとも呼ばれる)を表

す。該書式の該概観は、複数の列があり、境界が追加され、色が追加される等のように変更されてよい。このようなプレゼンテーション様式の改変は、大部分の任意のウェブオーサリングツールを用いて容易に実行することができる。HTMLおよびJavaScriptのさらに徹底的な理解を必要とする変更の例は、以下を含む。すなわち(1) フィールドを、チェックボックスなどのさまざまなUI構成要素に変更する、および(2) データベース照会に対処するために他のリポジトリへのリンクを組み込むなどの強化された機能を追加する。該クライアントは、書式を呼び出し、いくつかの書式フィールドはクライアントの詳細なビューから更新される。同様にフィールドからのユーザ入力が活動属性を更新する。

#### 【0123】

典型的なワークフローアプリケーションは、以下の機能を含んでよい。すなわち、サーバに接続する。プロセス定義(テンプレート)を構築する。作業リストを取得する。定義、あるいはインスタンスリストを処理する。プロセスの履歴を表示する。プロセスおよび活動のステータスを取得する。新規プロセスインスタンスを起動する。プロセスインスタンスを修正する。書式を定義し、プロセス定義またはインスタンスと関連付ける。添付物をプロセスインスタンスと関連付ける。および活動選択肢オプションを実行する。

#### 【0124】

該ワークフローシステムは、「ボックスの中からの」ウェブクライアントを提供する。これらの事前に定義されたクライアントは、すべてのタイプのユーザに、ユーザ関与および通知の変化する度合いを提供する。該ワークフローシステムはモジュラー式コンポーネント化クライアント構造を有する。すべてのクライアントは同じ構成要素の中から構築される。すべてのClient(クライアント)ユーザインタフェースは、いくつかの構成要素に基づいている。すべての構成要素は、該構成要素が混合され、整合されるようにHTMLおよびJavaScriptを使用していっしょに配列されるか、あるいはHTMLおよびJavaScriptを使用してクライアントから削除されさえる。

#### 【0125】



該クライアントの機能性は、使用量に応じて異なるように配列される。カスタムソリューションを必要とする組織の場合、該ワークフローシステムウェブクライアントは、JavaBean構成要素技術を活用し、該クライアントを一般的なサードパーティウェブ開発ツールのパレット内で容易に再構成、拡張またはカスタマイズできるようにする。該構成要素のそれぞれ一つは一定のインタフェースを実現する。考え方は、ユーザが、インタフェースが提示されている方法を好まない場合、該ユーザが独自のインタフェースを、該ユーザが好むやり方で構築することができるということである。たとえば、パッケージAppinterf a c e s は、サムネイルオーバービュー用のパブリックインタフェース(ThumbnailView)である。該ユーザがこのインタフェースを実現する限り、該ユーザは該ワークフローシステムサムネイルオーバービューを削除し、自分のものを同様に構成要素の残りとはプラグ接続することができる。

#### 【0126】

ThumbnailViewは、該計画(テンプレート)を特定できるインタフェースまたは現在表示されているプロセスインスタンスであるか、あるいは該計画(テンプレート)または表示されなければならないプロセスインスタンスを支配する。それは、活動およびプロセスインスタンスに対する選択更新をサポートする。まさにユーザが、該ワークフローシステムサムネイルオーバービューの代わりに独自のサムネイルオーバービューを使用できるように、該ユーザは、該ユーザが該オーガナイザパブリックインタフェースを実現する限り、該ワークフローシステムオーガナイザインタフェースの代わりに独自のオーガナイザインタフェースを使用することもできる。該インタフェースの該構成要素構造の背景にある重要な目的とは、ユーザが専用のインタフェースを指定クライアントに提供することができるという点である。

#### 【0127】

多様な構成要素は、別々のHTMLページである。したがって、それらは同じ機械の別個のブラウザの上で実行できる。それらが互いに対話し、互いについて知る方法は、アプレットの一つがロードされると、それが行う最初のことは、クライアントレジストリのようなapplistクラスを使った登録である。

該構成要素の一つが立ち上がると、それは該 `applist` クラスを使って登録し、それが、該 `applogic` がそれらについてどのようにして知るかである。Eメールクライアントからの形式の一つが選択され、該書式が別個の `html` ブラウザから起動されると、該書式は、それ自体を該 `applist` で登録する。その結果、該 `applist` は該 `applogic` に通知する。該 `applogic` は、ユーザのためにカレントコンテキストを追跡調査する。それは指定ユーザの書式および／またはプロセスを追跡調査し、その結果、それは、それが属性を取得するために移動しなければならないプロセスのインスタンスがどれかを知ることになる。

#### 【0128】

書式が選択されると、HTML形式は自動的に生成され、その中には機能が埋め込まれている。いったん書式が生成されると、ユーザがさらに何かを行わなくても、該サーバへの接続性が起こる。生成されたHTML書式は、スマート埋め込みアプレットを含む。任意の `html` エディタまたは任意のブラウザは、該書式機能性のどれかと干渉する `Java` アプレットなしにこの書式を立ち上げることができる。該HTML書式が立ち上げられると、該ワークフローシステムが、世界のどこかでいつの時点でだれによって何のデータが入力されているのかを知るように、該アプレットは、該ウェブサーバに移動し、プロセスに接続し、プロセスの内側の活動に接続することができる。

#### 【0129】

該ワークフローシステムが、ユーザ属性データを有する書式を生成すると、該ワークフローエンジンは、その活動へ接続されたプロセスに従う。ユーザが形式を立ち上げると必ず、サーバエンジンがこの書式のフィールドを使用できるようになり、万が一ユーザがデータを入力またはデータを変更すると、該サーバエンジンはそれを知ることになる。該サーバエンジンは該データを詰め込み、該ワークフローシステムは、誰が読取り／書込みを行ったのか、および誰が何を行ったのかを知っており、該ワークフローシステムはそのすべてを純粋な `html` を用いて行う。外部で発行された規格、すなわちHTMLを使用する該ワークフローシステムの優位点とは、ユーザが、アプリケーションを接続するために学習しなけ

ればならない独自ツールの内部をユーザが学習する必要なく機能が提供されるという点である。

#### 【0130】

該ワークフローシステムは、ユーザに、書式を続ける必要のある属性を指定させる。書式自体には、不可視の書式アプレットがある。該アプレットは、どのようにして該サーバに行き、それが形式フィールドを入力するために必要とする情報を主津臆するのかを理解することができる。そして、それが実行されると、それは該サーバに該情報を出力することができる。

#### 【0131】

該書式が表示されると、該書式の表示は該クライアントの残りの存在を仮定する。該書式が立ち上がると、それは該 `applist` とハンドシェーク／登録し、該 `appllogic` に対して生成される。 `Appllogic` は、書式のコンテキストを設定する。該書式は、それが、ネームタイプなどのどの種類の要素を表示するのかを知っており、該 `appllogic` は書式に、それが参照する、および該書式が適用する活動に関連するプロセスインスタンスを提供する。 `Appllogic` は、書式に `PIDx` (プロセス `id`) および `AIDx` (活動 `id`) を与える。該書式には `html` アドレスがあるが、ユーザがブラウザに移動し、クライアントオブジェクトなしに該書式から該 `URL` を選択する場合、該書式は何もできない。該薄いクライアントには、それが該プロセスインスタンスのコンテキストを有するように、該書式アプレットに必要なすべての機械が備えられている。

#### 【0132】

該ワークフローシステムは、以下の事前に定義されたクライアントを提供する。すなわち、 `Developer` (開発者) (`Fat` (太っている)、 `Full` (完全)) クライアント、 `Simple` (単純) クライアント、 `Instantiation` (`New Process`) (具体化 (新規プロセス)) クライアント、 `E-mail` (`Thin`) (Eメール (薄い)) クライアント、およびワークフロー管理者である。該ワークフロー管理者は、活動に対する許可を与え、該ワークフロー管理者ユーザタイプは、権限を有する誰かである。該 `Deve`

l o p e r (開発者) クライアントは、S i m p l e (単純) クライアント、I n s t a n t i a t i o n (具体化) クライアント、およびE - M a i l (Eメール) クライアントから構成されている。

【0133】

該D e v e l o p e r (開発者) クライアントは、すべての他のクライアントのすべての機能性を含む。あらゆるクライアント機能は、該D e v e l o p e r (開発者) クライアントによって提供される。該D e v e l o p e r (開発者) クライアントのサブセットは、他のクライアントによって提供される。該D e v e l o p e r (開発者) クライアントは、計画テンプレートを作成し、プロセスを具体化し、作業アイテムにアクセスし、応答するために使用される。該D e v e l o p e r (開発者) クライアントユーザタイプは、プロセスの設計、実現を担当するプロセス設計者または人員である。該S i m p l e (単純) クライアントは、作業アイテムにアクセスし、応答し、ときおりプロセスを具体化するために使用される。

【0134】

該S i m p l e (単純) クライアントにより、エンドユーザは、作業リストを受け取り、形式を記入し、該割り当てられた活動に関連付けられる添付物を表示／編集することができる。該S i m p l e (単純) クライアントユーザタイプは、プロセスを実現し、関与することを担当する人員である。該I n s t a n t i a t i o n (具体化) クライアントにより、その時々ユーザが、パブリックワークフロー計画をブラウズし、新規プロセスインスタンスを作成することができる。該I n s t a n t i a t i o n (具体化) クライアントユーザタイプは、新規プロセスの実現を担当する人員である。該Eメールクライアントが、作業アイテムに応えるために使用される。該Eメールクライアントは、単一の割り当てられた活動のビューを提供するURL起動可能クライアントである。該Eメールユーザタイプは、割り当てられたタスクだけを実行することを担当する数少ないユーザである。

【0135】

該ワークフローシステムは、「オンライン、リアルタイム」ユーザに協力ワー

クフローを提供する。該ワークフローシステムは、タスクまたはプロセスを完了するための協力作業を含む作業グループ関与のために設計されていた。このプロセスへの関与は、オンライン（速い）であり、リアルタイムでタスクに取り組む傾向がある。該ワークフローシステムは、二つの異なる種類のブラウザベースクライアント、すなわちDeveloper Client（開発者クライアント）とSimple Client（単純クライアント）を含むこれらのユーザに対処する。

【0136】

直感的なグラフィックデザイナーを含む該Developer（開発者）クライアントは、該ワークフローシステムでプロセス計画を定義するために使用される。図8は、ウェブベースのDeveloper Client（開発者クライアント）の画面表示を示す。該Developer Client（開発者クライアント）は、（活動、参加者、および形式などの）ワークフローの属性が定義できる豊富な開発フレームワークを提供する。

【0137】

いったん定義される（作成される）と、該計画は「プロセステンプレート」として記憶され、プロセスとして修正または開始できる。それはおもにプロセス定義のために使用されるが、該プロセスは実行中でさえ起動（開始）し、編集することができるため、該Developer Client（開発者クライアント）によって、ユーザはアクティブプロセスに関与することもできる。このようにして、該Developer Client（開発者クライアント）により、ユーザはオンラインでリアルタイムで作業できる（繋がれているユーザ）。該Developer Client（開発者クライアント）だけが、計画／テンプレートを定義および／または修正できる。

【0138】

該Developer Client（開発者クライアント）を介して、ウェブベースの形式は、単純な内部形式エディタ／生成器を使用して、容易に作成することができる。形式は、一般的なサードパーティウェブオーサリングツールの開発環境でもカスタマイズすることができる。該Developer Clie

nt (開発クライアント) は、以下を可能にする。すなわち、返却および貸し出しをサポートする添付物の表示および編集。添付物の「マスタコピー」の回覧一多くが読み取り、一つが書き込む。ポイントおよびクリック機能性による計画の作成。「変更およびマージ」の複数の添付物コピーの防止。および固定またはロックされていない柔軟なプロセス設計。

#### 【0139】

該Developer Client (開発者クライアント) は、複数の対話領域に分割される。図9は、該Developer (開発者) クライアントの画面レイアウトの画面領域および名称を示す。図9に図示されている画面レイアウトは、該Developer Client (開発者クライアント) にだけ適用可能である。

#### 【0140】

該Menu Bar (メニューバー) 500は、多くのアプリケーション全体機能を実行するためのメニューコマンドを提供する。該メニューアイテムは表1に示されている。該ツールバー502は、最も頻繁に使用されるメニューコマンドを実行するための迅速な代替作としてボタンを提供する。表2は、Tool bar (ツールバー) に表示されるボタンを示す。Status Bar (ステータスバー) 表示としても知られているAuxiliary Information Bar (補助情報バー) 504が、ヒントおよびステータス情報を手伝う。該ステータス情報は、該ワークフローシステムおよび/または該ワークフローシステム内のアイテムを提供する。

#### 【0141】

該Organizer (オーガナイザ) 506は、Organizer (オーガナイザ) UI構成要素内の選択更新にサポートを提供するオブジェクトのインタフェースである。該Organizer (オーガナイザ) 領域は、プロセステンプレート、プロセス、および活動のためのファイルキャビネットである。Organizer (オーガナイザ) を使用すると、該ユーザは、指定された基準を満たすそれらのアイテムだけが表示されるようにキャビネットの引出しをフィルタリングできる。

## 【0142】

アイテムは、ユーザにとって有意義である方法で並べ替えることができる。該ユーザがOrganizer（オーガナイザ）であるアイテムを選択すると、Overview Area（オーバビュー領域）およびTarget Area（ターゲット領域）にそのアイテムに関連する情報が入力される。該Organizer（オーガナイザ）は、多くの有効な詳細を視覚的に伝達するために色およびアイコンを使用する。該Organizer（オーガナイザ）内の活動は、該Simple Client（単純クライアント）内の活動と同じである。

## 【0143】

該Overview（オーバビュー）領域508は、該Organizer（オーガナイザ）内で選択されるプロセス／テンプレートのワークフローの粒状の縮小図（サムネイル）を表示し、Target Area（ターゲット領域）に全スケールを表示する。これにより、ユーザは、一覧活動タイプおよびプロセス内の活動の関係性を見ることができる。プロセスの場合、該ユーザは、該プロセス内の活動のステータスを一覧してみることもできる。サムネイルビューでは、色が活動の完了ステータスを特定する。

## 【0144】

該Target Area（ターゲット領域）510は、複数の目的に使用される。該Target Area（ターゲット領域）とは、ユーザが活動に応答する、またはプロセス／テンプレートの特徴を表示／修正するところである。この領域内に表示される情報は、ユーザが該Organizer（オーガナイザ）またはOverview Area（オーバビュー領域）で行う選択によって制御される。該ユーザは、属性およびワークフローなどの（Detailed View（詳細ビュー）としても知られている）活動に特殊な情報、および（Graphical View（グラフィックビュー）としても知られている）プロセス／テンプレート情報という2種類の情報を表示できる。

## 【0145】

活動に特殊な情報は、活動についての詳細を見るため、および活動に応答するために表示される。該ユーザは、Activity List（活動リスト）内

で活動を選択することにより活動情報を表示することができる。活動が選択されると、作業アイテムビューが該Target Area (ターゲット領域) に表示される。その作業アイテムビューは詳細なビューであり、e-メールクライアントの作業アイテムビューとまったく同じである。該詳細な作業アイテムビューは、表示のレイアウトに関係なく該作用アイテムの詳細なビューである。

【0146】

該要素は、同様である。該作業アイテムビューの要素はいくつかの活動属性およびプロセス属性、添付物リスト、形式、および添付物を追加、選択するのに役立つ制御である。該プロセス/テンプレートワークフローは、プロセス/テンプレート内の活動の関係性を図表で見えるため、および該プロセス/テンプレートの特徴を修正するために表示される。該ワークフローでの色の使用量は、該Organizer (オーガナイザ) と該Overview Area (オーバビュー領域) 内の色と同じ意味を持つ。

【0147】

該ユーザは、該Process/Template List (プロセス/テンプレートリスト) 内でプロセス/テンプレートを選択することによって、あるいは該Activity List (活動リスト) 内で活動を選択し、Graphical View (グラフィックビュー) をクリックすることによってワークフローを表示できる。該Target Area (ターゲット領域) のサイズを拡大する場合は、View (ビュー) メニューのHeighten Canvas (キャンバス強調表示) またはWiden Canvas (キャンバス拡大) メニューオプションを選択する。これで、スクロール可能な領域が拡大する。

【0148】

典型的な毎日のユーザが従事するだろう重要な活動とは、いったんプロセスが設計、実現されたら、プロセスに関与することである。ウェブスターの辞書は、プロセスを「所望される結果に向かう過程、ステップ、または方法」と定義している。プロセスは、該ワークフローシステム内で、フローチャート状の図でモデル化される。プロセスとは、実行されなければならないタスク (活動)、タスクが実行されるシーケンス、誰がタスクを実行できるのか、および該プロセスを完了



するために必要とされる人工遺物（文書および形式）の記述により定義される動的なエンティティである。すべての実行中のプロセスは、該Process List（プロセスリスト）に一覧表示される。

【0149】

柔軟性は、プロセスユーザに、タスクのための複数のオプションを提供することによってプロセスの中に作りこむことができる。活動が実行される準備が完了すると、それは、誰もがそのステータスを見ることができるよう、該Organizer（オーガナイザ）の中の該Universal Activity（To Do）（汎用活動（実行））リストに表示される。該Universal Activity List（汎用活動リスト）（またはさらに簡略には、該Activity List（活動リスト））は、すべてのプロセス内のすべての活動およびそのステータスを追跡調査する。図10は、該Universal Activity List（汎用活動リスト）を示す。発注要求は、該ワークフローシステムが理想的にモデル化するのに適する典型的なビジネスプロセスである。

【0150】

ユーザがプロセスを用いて実行することができる複数のことがある。ユーザは、プロセスのリストを並べ替え、フィルタリングし、プロセスのワークフローを表示し、プロセスのプロパティを表示し、文書をプロセスと関連付け、プロセスの履歴を表示することができる。並べ替えプロセスは、指定された順序でそれらを配列する。フィルタリングプロセスは、指定された特徴を満たすそれらのプロセスだけを表示する。プロセスを並べ替えるには、Sort（並べ替え）ドロップダウンリストをクリックし、所望の並べ替えオプションを選択する。プロセスをフィルタリングするには、Filter（フィルタ）ドロップダウンリストをクリックし、所望の特徴を選択する。「My... Processes（私の... プロセス）」は、ユーザが起動したそれらのプロセスを指す。「... Active Process（... アクティブプロセス）」は、アクティブ活動を含むそれらのプロセスを指す。

【0151】

プロセスを表示するために、該プロセスは、該Organizer（オーガナイザ）の該Process List（プロセスリスト）内で選択される。該プロセスの該ワークフローは、該Target Area（ターゲット領域）に表示され、同じワークフローのサムネイルビューは、該Overview Area（オーバビュー領域）に表示される。図11は、画面表示の右フレーム内のワークフロープロセスを示す。

#### 【0152】

いったんプロセスが表示されると、プロセスのプロパティが表示できる。Attachments（添付物）の例外により、プロセスプロパティは該プロセスのテンプレートから引き出される。添付物だけがプロセスに関して修正されてよい。添付物は、本項のつぎの部分に記述される。該Process（プロセス）メニューProperties（プロパティ）コマンドが、プロセスプロパティを表示するために選択される。

#### 【0153】

文書は、任意の時点でプロセスと関連付けられてよい。文書は任意のコンピュータファイルである。形式とは異なり、文書は、典型的には構造化されておらず、他のデータリポジトリとのリンクを持たない。文書がいったんプロセスと関連付けられると、それは添付物であり、すべてのプロセス参加者が利用できる。添付物のビューワは、該添付物を開くために適切なツールおよび／または環境セットを持たなければならない。文書は、テンプレートではなく、実行中のプロセスとだけ関連付けられてよい。添付物は活動からアクセスされるが、それらは特定の活動に結び付けられていない。文書をプロセスから切り離すためには、Remove（削除）をクリックする。これにより、文書自体ではなく、関連付けが削除されることになる。

#### 【0154】

プロセス履歴は、プロセスが開始された日付と時刻、ならびに誰によって、および活動が完了した日付と時刻、該活動の内部id、該活動に講じられる措置、および誰によってを含む。活動が、該ワークフローシステムによって自動的に実行された場合には、「プロセス」は、Responsible（担当）欄に表示

される。プロセス履歴を表示するには、ユーザは該Process List (プロセスリスト) 内で該プロセスを選択し、該Process (プロセス) メニューHistory (履歴) コマンドを選択する。

#### 【0155】

ユーザが活動によって実行できる複数のことがある。該ユーザは、活動のリストを並べ替え、フィルタリングし、活動記述を表示し、形式または添付物を開き、編集目的のために開かれた添付物を返し、添付物を開くために使用するアプリケーションを指定し、活動に応答することができる。活動の並べ替えは、指定順序で活動を配列する。フィルタリング活動は、指定された特徴を満たすそれらの活動だけを表示する。活動を並べ替えるためには、該ユーザはSort (並べ替え) 選択ボタンをクリックし、所望の並べ替えオプションを選択する。活動をフィルタリングする場合、ユーザはFilter (フィルタ) 選択ボタンをクリックし、所望の特徴を選択する。

#### 【0156】

活動詳細を表示するには、該ユーザは該Organizer (オーガナイザ) の該Activity List (活動リスト) の中の該活動を選択する。該Target Area (ターゲット領域) に示されている活動情報は、該活動の名前、この活動の責任者 (複数の人物に責任が譲渡されていることがある——この場合、譲受人ごとに該Activity List (活動リスト) 内に別個の活動がある)、該プロセスを開始した人物、および該活動の記述を含む。

#### 【0157】

形式を開く場合：

1. 該活動の詳細な記述がすでに表示されていない場合には、それを表示する。
2. Select Form (形式選択) (該当アイコン) をクリックし、開く形式を選択する。

#### 【0158】

フィールドに加えられる変更は、該形式が閉じられると、他のプロセス参加者が観察できる。

## 【0159】

添付物を開くには複数の方法がある。該ユーザは、それを開くために該添付物を作成するために使用されたアプリケーションを有していなければならない。加えて、アプリケーションファイル拡張子の関連付けが定義されなければならない。該ユーザは、それがすでに表示されていない場合には該活動の詳細な情報を表示し、Select Attachment (添付物選択) (該当アイコン) をクリックすることによって該活動の詳細なビューからだけ表示し、開く添付物を選択するために添付物を開いてよい。

## 【0160】

該ユーザは、(1) それがすでに表示されていない場合には、該プロセスのワークフローを表示すること、(2) 該Process (プロセス) メニューProperties (プロパティ) コマンドを選択すること、および(3) Attachments (添付物) タブを選択すること、(4) Current Attachments List (カレント添付物リスト) の中の該添付物を強調表示すること、(5) 表示するためだけに該添付物を開くためにView (ビュー) をクリックすることによって、該プロセスのグラフィックビューを表示したり、該プロセスのグラフィックビューを編集するために添付物を開いてよい。

## 【0161】

該添付物を開き、それに修正を加えるためにはEdit (編集) をクリックする。一度に一人の人物だけが添付物を編集してよい。しかしながら、任意の数の人が添付物を表示できる。該ユーザが編集のために添付物を開く場合、該ユーザは、他の人が、必要な場合にそれを編集できるように、自分が終わったらそれを返すことを覚えていなければならない。該ユーザは、変更が該添付物に反映されるためには、その元の名前に沿って添付物を保存しなければならない。

## 【0162】

該ユーザは編集のために添付物を開く場合、該ユーザがそれを元に返すまで、他の誰も該添付物を編集したり、該ユーザの変更を見ることはできない。添付物は、(1) Document (文書) メニューCheck in (返却) コマンドを選択すること、(2) 害添付物を返すために選択すること。Path (経路

）フィールドには、該添付物が記憶されるロケーションが入力され、変更されてはならない。（3）Check In（返却）をクリックすること、（4）Close（閉じる）をクリックすることによって返される。

#### 【0163】

該ワークフローシステムが、添付物を開くためにどのアプリケーションを使用すべきかを知るためには、アプリケーション-f i l e . t y p e 拡張子の関連付けが行われなければならない。文書拡張子は、「.」に続くファイル名内の文字である。指定された関連付けに基づき、該ワークフローシステムは、指定されたアプリケーションを用いて指定された拡張子のあるすべての文書を開くことになる。

#### 【0164】

関連付けは、（1）Document（文書）メニューAssociations（関連付け）コマンドを選択すること、（2）Type（タイプ）フィールドに該文書のタイプを入力すること（たとえば、Word）、（3）このタイプの文書のファイル名拡張子をExtフィールド（たとえば、doc）に入力すること。文書タイプに複数の考えられる拡張子がある場合、空間によって区切られている拡張子を入力する（たとえば、html）。（4）Program（プログラム）フィールドに入力するか、Browse（ブラウズ）をクリックし、この種の文書を開くために使用するプログラムのパス名全体を選択する。（5）Add（追加）をクリックすること。（6）Document Association（文書関連付け）ダイアログボックスを閉じるために、Close（閉じる）をクリックすることによって定義される。関連付けは、リストの中でそれを選択し、Remove（削除）を選択することによって削除される。

#### 【0165】

ユーザは、該活動に作用することによって活動割当てに応答してよい。これは、該プロセス状態を該プロセスの中のつぎの活動に進める。活動に作用することは、（1）該活動詳細情報を、それがすでに表示されていない場合には表示すること、（2）該活動を実行するために必要な処置を講じる。これは、添付物を該プロセスと関連付ける、あるいはすでに関連付けられた添付物を修正すること、

および(3)該講じられる処置を反映する該Target Area (ターゲット領域)の底部でボタンをクリックする(またはオプションを選択する)ことを含む。該プロセス内のつぎの論理活動が起動され、該Activity List (活動リスト)に表示される。

【0166】

該User Profile (ユーザプロファイル)は、該ユーザの個人的な好みを含む。該ユーザは、活動が自分に割り当てられたときに、自分がeメールによって通知されることを望むのかどうかを設定できる。加えて、該ユーザは、該(DMS)に該デフォルト保存ディレクトリを設定できる。これは、該ユーザが該文書を該(DMS)に追加するときはつねに該(DMS)が開かれるディレクトリである。

【0167】

Eメール通知は、(1)File (ファイル)メニューのUser Profile (ユーザプロファイル)コマンドを選択し、Notification (通知)チェックボックスを選択し、(3)該Eメールフィールドに完全なEメールアドレスを入力し、OKをクリックすることを含む。ユーザのデフォルト(DMS)ディレクトリのセットアップは、(1)File (ファイル)メニューのUser Profile (ユーザプロファイル)コマンドを選択し、(2)該Document Management (文書管理)タブを選択し、(3)該デフォルト(DMS)ディレクトリに関して入力するまたはBrowsing (ブラウズ)し、(4)OKをクリックすることを含む。

【0168】

色は、ステータスおよびタイプ情報を通信するために、該Organizer (オーガナイザ)、Overview Area (オーバビュー領域)およびTarget Area (ターゲット領域)内で使用される。活動の以下の色が、規定の意味を持つ。

【0169】

INSERT GRAPHIC (グラフィック挿入) 12 (青) -活動が完了した。

INSERT GRAPHIC (グラフィック挿入) 13 (明るい緑み青) - 活動は未決であるか、それはテンプレート活動であるかのどちらかである。

INSERT GRAPHIC (グラフィック挿入) 14 (緑) - 活動が現在の活動である。

【0170】

カラーシェードは、様々なコンピュータでわずかに異なってよい。

【0171】

特殊アイコンは、有効な情報を伝達するために、該Organizer (オーガナイザ) の該Process List (プロセスリスト)、Template List (テンプレートリスト)、およびActivity List (活動リスト) で使用される。表3は、特殊アイコンを示す。プロセスの設計および実現は、既存の手動プロセスに関する実質的な理解を持つ誰かによって実行されるだろう数少ない活動である。典型的には、人物は、どのようにすればプロセスを最良に設計し直せるかに関して、それが該ワークフローシステムによって実現されるときに最適化できるように考慮することになる。この活動は、知識作業によって協力して実行されてよい。

【0172】

テンプレートは再利用可能なプロセス定義である。それらには、プロセスが含むすべての情報が含まれるが、それらはアクティブではない。それらは、該プロセスの動作およびプロパティを定義するためにプロセスと関連付けられる静的なエンティティである。プロセスは、実行中テンプレートとして考えることができる。プロセスが作成されるとすぐに、それは実行中であり、(該フィルタがそれを排除するために設定されていないのであれば) 該プロセス内の第1活動が該Universal Activity List (汎用活動リスト) に表示される。該プロセスのテンプレートは、どの活動がつぎにアクティブになるのかを、過去の活動に対する応答に基づき定義する。プロセス/テンプレートは、二つの重要な特徴によって定位される。すなわち (1) 活動、および (2) フロー制御である。

【0173】

活動は、プロセスモデルのビルディングブロックである。それらは、該タスクと関連付けられている形式および該タスクを実行するために割り当てられている人員だけではなく、プロセス内の該タスクを定義する。活動が進行中であるとき、該活動はアクティブ状態にあり、該活動に割り当てられている人物がそれに応答するまで不活性にされないことになる。活動は、人間の行為または意思決定を必要とするタスクをモデル化するために使用される。活動は、図12に図示されるように、矩形ノードによってワークフロー内で図表を用いて表される。

#### 【0174】

活動のシーケンス/タイミングを制御する複数の異なるフローオペレータがいる。それぞれのオペレータは、別個の記号により図表で表される。矢印の例外はあるが、この記号はノードと呼ばれる。該オペレータは、表4に示されているものを含むが、それらに制限されない。その他の事前に定義されたノードは、Subplan (サブプラン) ノードおよびSplit Activity (活動分割) ノードを含む。

#### 【0175】

好ましい実施の態様においては、テンプレートは、(1) Graphical View (グラフィックビュー) の中にいる (すなわち、該Process List (プロセスリスト) またはTemplate List (テンプレートリスト) のどちらかが表示される)、(2) 該Template (テンプレート) メニューNew (新規) コマンドを選択する、(3) テンプレートのワークフローを定義する (該Target Area Toolbar (ターゲット領域ツールバー) が、該ワークフローが定義できるように表示される)、(4) テンプレートの一般記述属性を定義する、(5) このテンプレートから作成されたプロセスのオーナーを設定する、(6) 形式に利用可能なデータアイテム (ユーザ定義属性) を定義する、(7) 形式を作成し、それらを活動またはStart (開始) ノードと関連付ける、(8) 活動ノードの特徴を定義する、(9) あらゆるOR (論理和ノード) があるならば、OR (論理和) ノードの特徴を定義する、(10) あらゆるAND (論理積ノード) があるならば、AND (論理積) ノードの特徴を定義する、(11) あらゆる条件ノードがあるならば、該条件ノード



の特徴を定義する、(12) テンプレートを作成し、該Template (テンプレート) メニューSave (保存) コマンドを選択し、該テンプレートの名前および記述を入力することによって、最初から作成される。ステップ4-11は、ステップ6がステップ7と11の前に実行される限り、任意の順序で実行されてよい。

#### 【0176】

テンプレート作成者は、任意のテンプレートプロパティを設定、修正してよい。しかしながら、該テンプレート作成者は、いったん該テンプレートがプロセスを開始するために使用されると、テンプレート特性を修正してはならない。加えて、該テンプレート作成者は、誰か他の人によってすでに修正されているテンプレートを修正してはならない。既存のテンプレートに変更を加えるには、該Template (テンプレート) メニューEdit (編集) コマンドを選択することによって該編集モードに入り、変更が加えられ、該テンプレートが保存される。プロセスの該プロパティは、そのテンプレートから引き出されるが、修正されてはならない。以下のプロパティは、プロセスおよびテンプレートに適用する。

#### 【0177】

- ・ワークフロー — 該テンプレート／プロセス内の活動のシーケンス
- ・一般属性 — 該テンプレート／プロセスの名前、題名、および記述
- ・オーナー — (テンプレートだけに適用する) — テンプレートの該オーナー。

デフォルトによって、プロセスの該オーナーは、該プロセスが作成された元の該テンプレートの該オーナーである。

- ・データアイテム — 形式で表示できるデータ
- ・活動ノード属性 — 該活動の記述および割当て
- ・OR (論理和) ノード属性 — OR (論理和) ノードの記述および動作
- ・AND (論理積) ノード属性 — AND (論理積) ノードの記述および動作
- ・条件ノード属性 — 条件ノードの記述および動作
- ・形式 — 活動を完了するために使用される形式。

#### 【0178】

ワークフローを修正するためには、該テンプレートの該ワークフローが表示さ

れなければならない。該Target Area Toolbar (ターゲット領域ツールバー) は、ノードおよび矢印を設置するために使用され、図13に示される。ノードを追加するには(1) 該Target Area Toolbar (ターゲット領域ツールバー) が見えない場合には、該Template (テンプレート) メニューEdit (編集) コマンドを選択する。(2) 該Toolbar (ツールバー) の中で、追加されなければならない該ノードをクリックする。(3) 該ノードが設置されなければならない該Target Area (ターゲット領域) 内の空白領域をクリックする。(4) 設置されるノードごとにステップ3を繰り返す。

#### 【0179】

矢印を追加するには、(1) 該Target Area Node Toolbar (ターゲット領域ノードツールバー) が見えない場合には、該Template (テンプレート) メニューEdit (編集) コマンドを選択する。(2) 該Toolbar (ツールバー) の中で矢印演算子 (該当アイコン) をクリックする。(3) 矢印がそこから発する該ノードの非フィールド領域をクリックし、マウスボタンを放さないで、該矢印の端点を該宛先ノードまでドラッグする。矢印は二つのノードを繋がないといけない。二つのノードを繋がない矢印は描画できない。(4) 矢印の名前フィールドの中をクリックし、それに、矢印が取られることになるユーザ動作を記述する一意の名前を与える。活動ノードから発する矢印だけが、記述名を必要とする。それ以外のノードタイプはユーザ動作を必要としない。(5) 設置される矢印ごとにステップ3を繰り返す。

#### 【0180】

ノードまたは矢印を移動するには、該Node Toolbar (ノードツールバー) のボタンのノードが押されたままで、該ノードまたは矢印をクリックされ、所望された場所までドラッグされる。ノードを移動するとき、必ず非フィールド領域内でクリックする。ノードをドラッグするとき、接続されている矢印は自動的に調整し直されることになる。ユーザは、どちらの端点もノードに繋がれないように、矢印をドラッグしてはならない。

#### 【0181】

テンプレートの一般属性は、(1) 該テンプレートの該ワークフローを表示する、(2) 該Template (テンプレート) メニューProperties (プロパティ) コマンドを選択する、(3) 表示されているProperties (プロパティ) ダイアログボックスのGeneral (一般) タブの上で該テンプレートのName (名前)、Title (題名)、およびDescription (記述) を入力する、(4) OKをクリックすることによって定義される。テンプレートのオーナーの設定は、(1) 該テンプレートの該ワークフローを表示する、(2) 該Template (テンプレート) メニューProperties (プロパティ) コマンドを選択する、(3) 該Owner (オーナー) タブを選択する、(4) 該Role (役割) チェックボックスを選択し、クリックして該所望の役割を選択し、OKをクリックすることによって達成される。

#### 【0182】

役割とは、NTおよびExchange (交換機) 内のRecipient (受取人) コンテナ内のグループに同等である。グループ/コンテナの作成方法の詳細については、該Directory Service (ディレクトリサービス) 文書を参照しなければならない。使用可能な役割の該リストは、該Directory Service (ディレクトリサービス) 内に定義されているすべてのグループ/コンテナである。役割が選択されると、該役割のメンバーが右側ペインに表示される。

#### 【0183】

データアイテムは、ユーザが、形式をまとうことによって追跡調査することを希望するプロセスに関連付けられた属性である。たとえば、購入要求プロセスは、属性費用を有することがある。プロセス参加者は、費用の金額を設定するために該形式を開くことができる。

#### 【0184】

データアイテムのセットを指定するには、(1) 該テンプレートの該ワークフローを表示する。(2) 該Template (テンプレート) メニューProperties (プロパティ) コマンドを選択する。(3) 該User Defined Attributes (ユーザ定義属性) タブを選択する。(4) Ad

d (追加) をクリックする。(5) 該データアイテムの名前を修正する。(6) 該データアイテムの該データタイプを選択する。(7) このデータアイテムを含む形式が表示されると、このデータアイテムのフィールドが自動的に記入されなければならない場合に、該データアイテムの初期デフォルト値を入力する。(8) データアイテムごとにステップ4から7を繰り返す。(9) 修正された最後のフィールドの場合、変更が受け入れられるように、<Enter>キーを押すか、別のフィールドの中をクリックする。(10) OKをクリックする。

#### 【0185】

スクリプトは、(1) 該テンプレート／プロセスの該ワークフローが表示されることを確実にし、編集モードにある、(2) 該テンプレートのプロパティを表示するために任意のノード上でProperties (プロパティ) ボタン (該当アイコン) をクリックする、(3) 該Scripts (スクリプト) タブを選択する、(4) Init Script (スクリプト) フィールドでは、プロセスが最初にこのテンプレートから作成されると、実行するために該JavaScriptを入力することによって定義される、(5) 該Commit Script (コミットスクリプト) フィールドでは、該プロセス完了時に実行するために該JavaScriptを入力する、(6) OKをクリックする、(7) 修正終了時に、該Template (テンプレート) メニューSave (保存) コマンドを選択することによって定義される。標準ECMA JavaScript コマンドに加えて、該ワークフローシステムは、該Server (サーバ) から情報を検索するために、JavaScriptコマンドの追加のセットを提供する。

#### 【0186】

活動ノード属性は、(1) 該テンプレートの該ワークフローが表示されることを確実にする、(2) 該活動ノードの該一番上のフィールド内の活動に関してAssignee (譲受人) (Role (役割)) を入力する。代わりに、該譲受人は、以下のステップ6に説明されるように設定できることによって定義される。該Assignee (譲受人) は、該Directory Service (ディレクトリサービス) で定義されるグループでなければならない。役割は、E

xchange (交換機) 内のNTおよびRecipient (受取人) コンテナの中のグループに同等である。Directory Service (ディレクトリサービス) 文書は、グループ/コンテナを作成する方法に関する詳細について参照されなければならない。(3) 該活動ノードの底部フィールドに該活動の名前を入力する、(4) 該活動ノードで該Properties (プロパティ) ボタン (該当アイコン) をクリックする。

【0187】

(5) General (一般) タブでは、該活動の該Name (名前)、Title (題名)、およびDescription (記述) を入力する、(6) 該活動の該Assignee (譲受人) が前記ステップ2で設定されない場合、該Assignee (譲受人) タブをクリックし、該Assignee (譲受人) チェックボックスを選択し、所望の役割を選択し、OKをクリックする。利用可能な役割のリストは、該Directory Service (ディレクトリサービス) に定義されるすべてのグループ/コンテナである。役割が選択されると、該役割の該メンバーが右側ペインに表示される。(7) 該Script (スクリプト) タブでは、該活動がアクティブになるときに実行するためにBodyScript (ボディスクリプト) を、ユーザが該活動OR (論理和) クリックに応答するときに実行するためにEpilogue Script (エピローグスクリプト) を入力し・・・該(DMS)に記憶されているスクリプトを選択する、(8) OKをクリックすることによって定義される。

【0188】

OR (論理和) ノード属性は、(1) 該テンプレートの該ワークフローが表示されていることを確認する、(2) OR (論理和) ノード (該当アイコン) の該Properties (プロパティ) ボタン (該当アイコン) をクリックする、(3) 該Properties (プロパティ) タブを選択し、該ノードのName (名前) およびDescription (記述) を指定する、(4) 該Script (スクリプト) タブを選択し、該OR (論理和) ノードが完了するときに実行されるスクリプトを入力する、(5) OKをクリックすることによって定義される。AND (論理積) ノード属性は、(1) 該テンプレートの該ワークフロ

一が表示されることを確認する、(2) 該AND (論理積) ノード (該当アイコン) の該Properties (プロパティ) ボタン (該当アイコン) をクリックする、(3) 該Properties (プロパティ) タブを選択し、該ノードのName (名前) およびDescription (記述) を指定する、(4) 該Script (スクリプト) タブを選択し、該AND (論理積) ノードが完了すると実行される該スクリプトを入力する、(5) OKをクリックすることによって定義される。

#### 【0189】

条件ノード属性は、(1) 該テンプレートの該ワークフローが表示されていることを確認する、(2) 該条件ノード (該当アイコン) で該Properties (プロパティ) ボタン (該当アイコン) をクリックする、(3) Decisions (決定) タブを選択する、(4) 評価されるデータアイテムを選択する。該データアイテムオプションとは、該Template Properties (テンプレートプロパティ) ダイアログボックスの該User Defined Attributes (ユーザ定義属性) ページに指定されたものである。(5) 該条件ノードから発する矢印ごとに、該データアイテムを、該矢印が取られるために真でなければならない該関係に比較され、該関係を選択する、(6) デフォルト矢印を特定する該ラジオボタンを選択する、(7) 式のどれも真ではない場合にこの矢印が取られることになる。(8) 該リストの中の表記の順序は、該式を強調表示し、Up (アップ) またはDown (ダウン) をクリックすることによって配列し直すことができる。該条件ノードが起動または完了時にスクリプトを実行するには、該Script (スクリプト) タブが選択され、該スクリプトはそれぞれ該PrologフィールドまたはEpilogフィールドに入力される。(9) OKをクリックする。

#### 【0190】

形式は、該ワークフローシステムを使用して定義、生成される。それらは、そのレイアウトが、HTMLオーサリングツールを使用して、あるいはHTMLコード自体を修正することによって修正されてよい構造化されたフィールドベースのHTMLファイルである。それらは、該ワークフローシステムと構造化された

データリポジトリ間のデータ交換用人工遺物の役割を果たす。形式は作成され、活動またはテンプレート作成中のStart（開始）ノードと関連付けられる。これにより、それらをプロセス参加者が利用できるようになる。

#### 【0191】

形式は、（１）それがまだ実行されていない場合には、該テンプレートの該データアイテムを定義する、（２）該テンプレートの該ワークフローが表示されていることを確認する、（３）形式が関連付けられなければならない該活動ノードで該Form（形式）ボタン（該当アイコン）をクリックする。新規形式は、該Start（開始）ノードから作成することはできない。既存の形式だけが、該Start（開始）ノードと関連付けることができる、（４）Create New（新規作成）をクリックする。該形式の内容を指定するためのダイアログボックスが表示される。デフォルトによって、すべてのユーザ定義データアイテムが形式に表示されることになる。

#### 【0192】

（５）データアイテムごとに、ユーザが、データアイテムが該形式に表示されるのを望まない場合、該ユーザは、該データアイテムのUse in Form（形式内での使用）チェックボックスを解除する。それ以外の場合、該ユーザが該データアイテムの形式ラベルを指定する。最後のラベルが修正された後、該ユーザは、変更が受け入れられるように、<Enter>キーを押すか、別のフィールドの中をクリックするかのどちらかである。（６）該形式に（該DetailView（詳細ビュー）から該形式を開くために該プロセス参加者によって使用されることになる）該ファイル名よりさらに有意義な名前が与えられなければならない場合には、ユーザは該Form Title（形式題名）フィールドにその名前を入力する。該Form Title（形式題名）フィールドが記入されない場合には、該形式のファイル名が該形式の名前に関して使用されることになる。

#### 【0193】

（７）Form Path（該形式経路）フィールドの中にForms Repositoryの経路、およびDocument Management Sy

system (文書管理システム) に該形式を保存するためのファイル名を指定する。代わりに、該形式を格納するための場所に関してツリー構造をブラウズすることができる。Exchange (交換機) が該Directory Service (ディレクトリサービス) として使用されている場合には、該形式はFormsRepository 経路に保存できないことになる。このケースでは、該形式は、表示される上部レベルフォルダに対して保存される。プロセス参加者が該形式を開くことができる前に、Exchange Client (交換機クライアント) は、該Exchange (交換機) ディレクトリから前記に指定されたFormsRepository 経路に手動で該形式をコピーするために使用されなければならない。(8) Generate Form (形式生成) をクリックする。該形式は、該フィールドが単一欄にあるように作成される。該形式の該概観は、任意のHTML Editor (HTMLエディタ) を使用し、OK をクリックしてカスタマイズすることができる。

#### 【0194】

形式は、(1) 該テンプレートの該ワークフローが表示されていることを確認する、(2) 形式を関連付ける活動ノードで該Form (形式) ボタン (該当アイコン) をクリックする。しかしながら、Start (開始) ノードの例外がある。該Start (開始) ノードの場合、該ノードが選択され、該Activity (活動) メニューProperties (プロパティ) コマンドが選択される。新規プロセスが開始されると、該プロセス開始プログラムが該形式を記入できるように、該形式はただちに開かれることになる。(3) Browse (ブラウズ) をクリックし、該形式に関して該Document Management System (文書管理システム) をブラウズする。(4) 該Path (経路) フィールドに該形式のロケーションを入力する。代わりに、該ツリーは、該形式ロケーションまで横断できる。

#### 【0195】

(5) 該書式に該ファイル名より (該DetailView (詳細ビュー) から該形式を開くためにプロセス参加者によって使用されることになる) さらに有意義な名前が与えられなければならない場合には、その名前が該Title (題



名) フィールドに入力される。(6) エントリを受け入れるためにOKをクリックし、該Browse (ブラウズ) ダイアログボックスを閉じる。(7) 新しい関連付けを受け入れるためにOKをクリックし、該Activity Instance Properties (活動インスタンスプロパティ) ダイアログボックスを閉じる。書式を活動または該Start (開始) ノードから書式を切り離すには、Remove (削除) をクリックする。これで、書式自体ではなく、関連付けが削除されることになる。

#### 【0196】

新しいプロセスは、作成され、(1) プロセステンプレートのリストを表示するために、該Organizer (オーガナイザ) の該Template (テンプレート) をクリックする、(2) 該新規プロセスに使用される該テンプレートを選択し、該Toolbar (ツールバー) のNew Process (新規プロセス) (該当アイコン) をクリックする。(3) 書式が該プロセスのテンプレートで該Start (開始) ノードと関連付けられていた場合、該書式は、該書式を記入し、提出できる書式が開く。該新規プロセスは、該Process (プロセス) タブを選択し、該新規プロセスを強調表示することによって見ることができる。該新規プロセスは、該テンプレートと同じ名前を持つが、それを排除するためにフィルタがセットされていない限り、見られない。該新規プロセスは、該Target Area (ターゲット領域) 内に表示される。該プロセスでの該カレントアクティブ活動は、緑色っぽい色のノードによって強調表示され、(フィルタがそれらを排除するために設定されていないならば) 該Activity (活動) タブが選択されるときに、該Activity List (活動リスト) の中に表示されることになる。

#### 【0197】

プロセスに頻繁に参加するが、プロセスを定義しないユーザの場合、該ワークフローシステムはSimple Client (単純クライアント) を提供する。図14は、ウェブベースのSimple (単純) クライアントの画面表示を示す。該Simple Client (単純クライアント) は、オンラインで、およびリアルタイムでタスクに参加するユーザ向けに設計された。該Simple

Client (単純クライアント) は、頻繁なリアルタイム、オンラインタスク/活動参加である。該Eメールクライアントおよび具体化クライアントは該Simple (単純) クライアントの構成要素である。

【0198】

該Simple Client (単純クライアント) は、E-mail Client (Eメールクライアント) およびInstantiation Client (具体化クライアント) で利用できる機能を結び付ける。該Simple Client (単純クライアント) によって、ユーザは、ユーザがプロセスを具体化 (開始) し、ユーザは作業リストからその割り当てられたタスクを処理するので、文書を表示、編集、追加、および添付できるようにすることによってプロセスに参加できるようにする。該単純クライアント (Simple Client) は、作業アイテムにアクセスし、余分な情報を苦にする必要なく、新規プロセスを開始する単純な方法を提供する。

【0199】

該単純なクライアントは、基本的に二つの要素、すなわち二つのフレームである。図15は、該Simple Client (単純クライアント) ビューを二つの要素フレームとして示す。フレームとは、html ページが表示できる場所のことである。左側フレームは作業リスト、該ユーザに割り当てられる基本的な活動である。該右側フレームは、該Eメールクライアントを含む。このようにして、該単純クライアントは、ユーザに割り当てられているが、依然として完了していない活動を指す作業リスト (左側フレーム) とともに、詳細なビュー (右側フレーム) から構成されている。

【0200】

該Simple Client (単純クライアント) は、(図6に図示されていた) 完全クライアントの以下の要素を含む。すなわちApplication Logic (アプリケーション論理) (AppLogic) 66、Workflow Manager Model (ワークフロー管理モデル) 68、Form Model (形式モデル) 70、Workitem Model (作業アイテムモデル) 71、Process Model (プロセスモデル) 72

、およびTemplate Model (テンプレートモデル) 74を含む。視覚アプレットは、作業リストである。

#### 【0201】

ユーザが該URLをダブルクリックすると、該ワークフローシステムは、パスワードのログインに対する要求とともに戻る。アイコンが該作業リストから選択されると、それはウェブページを該詳細なビューのフレームの中に呼び出す。EメールクライアントのURLを選択する方法は、該URLがeメールの内側でハード符号化され、ユーザがその上でダブルクリップすると、該ウェブページが起動されるという事実を使用する。単純クライアントのケースでは、ユーザが活動リストから要素を選択すると、効果は、特定の活動に対する特定のアイテムを含む詳細なビューウェブページが起動されるという点である。

#### 【0202】

該Simple Client (単純クライアント) を呼び出すには、該Simple Client (単純クライアント) のURLが、ウェブブラウザ、たとえば「http://<i-Flow Server host name>/<http document directory, which is twf30 if default accepted>/twfsimple\_en.html?uid=<username>%server=<Server name>」に入力される。

#### 【0203】

該Simple Client (単純クライアント) -インタフェースは、以下の二つの追加を伴いE-mail Client (Eメールクライアント) と同じインタフェースを使用する。すなわち、(1) 作業アイテムリスト、および(2) 該New Process (新規プロセス) ボタンである。検討される、あるいは取り組まなければならない作業アイテムは該作業アイテムリストから選択される。該作業アイテム情報は、該E-mail Client (Eメールクライアント) の場合と同じように正確に提示される。該E-mail Client (Eメールクライアント) で利用できるすべての機能が、該Simple Client (単純クライアント) で利用できる。該NewProce

ss (新規プロセス) をクリックすると、新規プロセスを開始するために、該 Instantiation Client (具体化クライアント) が呼び出されることになる。

#### 【0204】

該EメールクライアントはHTMLとJavaScriptの両方から構築される。図6では、該Applogic (アプリケーション論理)、該WFManager Model (ワークフロー管理モデル)、該Workitem Model (作業アイテムモデル)、該Form Model (書式モデル)、該Process Model (プロセスモデル) および該Template Model (テンプレートモデル) が、単純なクライアントの一部である。視覚アプレットは、該作業リストである。

#### 【0205】

該ワークフローシステムは、購入要求、費用報告、訪問追跡調査、またはウェブコマースなどの内部管理プロセスにとっては理想的である。プロセステンプレートは、非技術人員によって作成され、特別様式で実行中に変更することができ、自動化ワークフロープロセスが実際のプロセスとともに進化できるようにする。

#### 【0206】

管理タスクは、多くの場合、多岐に渡るユーザによって時折より多く完了される。これらのユーザは、(「不定期の」ユーザとも呼ばれている) おそらくモデムを通した遠隔低帯域幅接続を有するもの、およびおもにeメールを通して作業するもの(繋がれていないユーザ)を含む。これらの種類のユーザを収容するために、該ワークフローシステムは二つの使用しやすいウェブベースのクライアント、すなわちInstantiation Client (具体化クライアント) およびE-mail Client (Eメールクライアント) を提供する。

#### 【0207】

(該Developer Client (開発者クライアント) を使用して) いったんプロセステンプレートが作成されると、不定期ユーザは、Instant

ti ation Client (具体化クライアント) またはE-mail Client (Eメールクライアント) という該二つのブラウザをベースにしたクライアントのどちらかを通して参加することができる。該InstantiationClient (具体化クライアント) と該E-mail Client (Eメールクライアント) の両方とも主に遠隔ユーザ向けのHTMLクライアントである。

【0208】

該InstantiationClient (具体化クライアント) は、eメール専用ユーザに、確立されたURLアドレスを通して所定のテンプレートから新規プロセスを具体化(起動) するための方法を提供する。該E-mail Client (Eメールクライアント) は、繋がれていないユーザが、その作業アイテムおよび関連文書を受け取ることができるようにし、彼らが、eメールを介して(SMTPまたは無線SMSを通して) そのタスクを、すべてウェブクライアントを通して完了できるようにする。このようにして、遠隔ユーザは、プロセス、および添付文書および書式の表示および編集に参加することができる。参加は、匿名の外部ユーザによる。

【0209】

これらのクライアントを用いて、タスクは割り当て、参加者に直接的に送り、参加者がオンライン作業リストを通して作業割当てをチェックする必要性を排除する。さらに、該ワークフローシステムの直感的なブラウザベースのクライアントを用いて、該インタフェースが単純で学びやすいため、タスクは、より効率的にたまの参加者によって完了することができる。

【0210】

該InstantiationClient (具体化クライアント) は、余分な情報を苦にする必要なく、新規プロセスを迅速に開始する(たとえば、休暇要求を提出する) 単純な方法を提供する。該具体化クライアントは、テンプレートからのリストから成り立っている。右側には、活動ではなく、テンプレートの詳細なビューがある。該詳細なビューは、新しい作成済みプロセスの初期条件を設定するのに役立つ。プロセスの開始時に追加される添付物は初期化されてよい。書式

は、完了されなければならない。開始する、または取り消すための一つのボタンがある。いったんユーザがプロセスを開始するのがOKであると言うと、プロセスの新規インスタンスは、それに追加された添付物で作成され、該プロセス属性は、書式に入力された値に従って初期化される。

#### 【0211】

該InstantiationClient (具体化クライアント) を呼び出すため、該InstantiationClient (具体化クライアント) のURLが、ウェブブラウザ、たとえば「http://<i-Flow Server host name>/<http document directory, which is twf30 if default accepted>/twfinstantiation\_en.html?uid=<username>%server=<Server Name>」に入力される。

#### 【0212】

該InstantiationClient (具体化クライアント) にログイン時にユーザが実行してよい唯一の動作は、新規プロセスの開始である。これには、該プロセスの該テンプレートがすでに存在していることが必要になる。該InstantiationClient (具体化クライアント) にボタンがない場合には、テンプレートは存在しない。新規プロセスは、(1) 具体化する該テンプレートのボタンをクリックする。該テンプレートの該名前および記述は右側ペインに表示されることになる。(2) 指定されたテンプレートのあるプロセスを開始するためにOKをクリックする。形式が該プロセスのテンプレート内のStart (開始) ノードと関連付けられていた場合、該書式が開くことになる。

#### 【0213】

(3) 該書式を記入し、それを提出する。該プロセスが開始され、該Developer Client (開発者クライアント) を使用して表示できる。該アクティブな活動は、該Full Client (完全クライアント) とSimple Client (単純クライアント) の両方で見ることができる。該プロセス内でのアクティブな活動の該譲受人がe-メール通知を受け取ることを選ぶと

、彼女／彼は該E-mail Client (E-メールクライアント) を介して該活動にアクセスできる。

【0214】

該E-mail Client (E-メールクライアント) は、基本的には、それがそこから起動できるアドレスを有するウェブページである。該E-mail Client (E-メールクライアント) は、HTMLとJavaScriptの両方から構築される。該E-mail Client (E-メールクライアント) は、クライアントの中で最小であり、それは余分な情報を苦にする必要なく、タスク割当てに迅速に応答する簡略な方法を提供する。該E-mail Client (E-メールクライアント) は、該ユーザに割り当てられているタスクの作業アイテムビューを提供する。

【0215】

それは、記述、日付、および時間等のようなアイテムを事前に定義した。該E-mail Client (E-メールクライアント) は、添付物および形式などのそのタスクに特殊なアイテムを有する。該E-mail Client (E-メールクライアント) は、タスクの完了後に、選択肢のセットから下されなければならない最終的な決定を有する。該E-mail Client (E-メールクライアント) にアクセスするためには、ユーザは、該ワークフローシステムからe-メール通知を受け取らなければならない。通知を受け取るために、該ユーザは、該ユーザプロフィールをセットアップするときに該Developer Client (開発者クライアント) 内でこのオプションを選択したに違いない。それから、ユーザは、タスクが該ユーザに割り当てられたときにe-メールを受信するに過ぎない。

【0216】

該e-メールメッセージの本文内では、クリックされると、該E-mail Client (E-メールクライアント) を呼び出すHTTPリンクがある。該E-メールクライアントは、HTMLページによって囲まれている。該E-メールクライアントは、パラメータのリスト (ヌルリストが一つのオプションである) を含むURLによって呼び出される。たとえば、「http://iflow/

TW30/thinclient」は、パラメータなしで該Eメールクライアントを呼び出すためのURLである。パラメータを指定して該Eメールクライアントを呼び出す場合、該パラメータ?PID=\_\_&AID=\_\_が、該プロセス識別を表す該URLコマンド行(PID)および該活動識別を表すAIDに追加されることになる。これが、該Eメールクライアントにパラメータを渡し、該Eメールクライアントページを立ち上げる方法である。このhttpは、eメールメッセージで書き込まれることになる。

#### 【0217】

該Eメールクライアントは、それが、該活動が完了されているときに該サーバに直接的に接続されるという意味でリアルタイムである。該E-Mail Client (Eメールクライアント)は、完了した活動を処理するために該蓄積交換パラダイムを使用しない。該E-Mail Client (Eメールクライアント)は、該ユーザがある特定の活動へのアクセスを取得するために該URLを送達するが、該活動が立ち上がると、それはオンラインで該サーバに接続される。オンラインとは、該eメールクライアントと該サーバの間の対話性が、リアルタイムでCORBAを介して実行されることを意味する。それは、ストアアンドフォワード パラダイムを使わない。これは、ユーザがeメールクライアントで完了ボタンを押し、それがオフラインであるときには、該情報が採取され、eメールメッセージの中に入れられ、該サーバに送信され、最終的に該サーバがそれを取得するシステムと対照的である。該ワークフローシステムでは、ユーザがOKを押すと、それはリアルタイムで該サーバに移動する。

#### 【0218】

該Eメールクライアントインタフェースは、該Developer Client (開発者クライアント)と異なって見えるが、その後ろにある機能性は、本質的には、該Developer Client (開発者クライアント)のサブセットである。該E-mail Client (Eメールクライアント)を用いて、ログイン時にユーザは、活動に応答し、文書をプロセスと関連付け、該活動と関連付けられている書式または添付物を開き、添付物を開くために使用する該アプリケーションを指定することができる。図16は、該Eメールクライア



ントインタフェースを示す。

【0219】

活動は、必要な書式および添付物を修正し、該活動に関して講じる処置を表す該ボタンをクリックすることによって応答される。活動に関連付けられている書式ごとに別個のボタンがある。ボタンをクリックすると、活動の対応する書式が開く。該活動に関連付けられた添付物ごとに別個のボタンがある。新しい添付物は、活動に追加できる。添付物が開かれる前に、（該ワークフローシステムDeveloper Client（開発者クライアント）ですで行なわれていない場合には）それを開くために使用する該アプリケーションが指定されなければならない。

【0220】

添付物は、該添付物に対応する該ボタンをクリックすることによって開かれる。該添付物は、表示専用あるいは他の誰も修正のためにそれを開かなかったのであれば修正用に開かれてよい。該添付物が開かない場合、それは、それを開くための該アプリケーションが指定されていなかったためである可能性がある。アプリケーションは、（1）Associations（関連付け）をクリックする、（2）Type（タイプ）フィールド（たとえば、Word）に該文書の種類を入力する、（3）この種の文書のファイル名拡張子をExtフィールド（たとえば、doc）に入力する。文書タイプが複数の考えられる拡張子を有する場合、空間で区切られた拡張子（たとえば、html）を入力する。（4）この種の文書を開くために使用する該プログラムの経路名全体を、該Program（プログラム）フィールドに入力する、またはBrowse（ブラウズ）をクリックして選択する、（5）Add（追加）をクリックする、（6）該Document Association（文書関連付け）ダイアログボックスを開じるためにClose（閉じる）をクリックすることによって添付物を開くために指定することができる。関連付けは、該リスト内でそれを選択し、Remove（削除）をクリックすることによって削除される。

【0221】

該ワークフローシステムSoftware Development Kit

(ソフトウェア開発キット) (SDK) は、該ユーザがそれらを活用できるように、該モデルクラスに該インタフェースを提供する。また、該SDKは、該開発者が、該開発者が有していることがあるどのようなアプリケーションともそれらを統合できるようにするために該UI構成要素のそれぞれの仕様を提供する。該UI構成要素およびモデルクラスに加えて、該SDKは、該スクリプトアダプタ、該DDフレームワーク、Directory (ディレクトリ) アダプタ、および該(DMS)アダプタのインタフェース定義を提供する。該SDKは、該Script (スクリプト) アダプタ、DDフレームワーク、Directory (ディレクトリ) アダプタ、および(DMS)アダプタの該開発者インプリメンテーションを提供する。該(DMS)アダプタは、Native (ネイティブ) NT、Exchange (交換機) およびJDKという三つのインプリメンテーションを有する。該Directory (ディレクトリ) アダプタは、Native (ネイティブ) NT、Exchange (交換機) およびLDAPという三つのインプリメンテーションも有する。

#### 【0222】

該SDKの一部は、ユーザが、該サーバにアクセスするスクリプトコマンドを書き込むことができるようにする該JavaScriptに追加された拡張子である。該スクリプトコマンドは、彼らが、該プロセスコンテキストにアクセスできるように、該ユーザに事前に定義されている。

#### 【0223】

該ワークフローシステムは、該開発者に、該開発者が専用のコマンド拡張子を追加できるようにするためのフレームワークを提供する。該開発者は、該ワークフローシステムの外で任意のデータリソースまたはアプリケーションにアクセスするJavaクラスを開発することができ、該開発者は該スクリプト機能環境の中からそのJavaクラスにアクセスできる。該スクリプトへの該インタフェースによって、該開発者は、異なる種類のスクリプトアダプタおよび異なる種類のスクリプト機能言語を組み込むことができる。

#### 【0224】

該ワークフローシステムSDKは、使用されている該アダプタとは無関係であ

り、このようにしてクライアントを開発するために該SDKを使用するための該 Value Added Retailers (付加価値小売業者) (VAR S) を可能にする。該SDKによって、開発者は、該ワークフローシステムをサードパーティの製品およびツールに統合するためだけではなく、ワークフローシステム機能性をカスタマイズし、拡張することもできる。該SDKはJava Beanをベースにしている。該SDKは、Java Bean およびJava Script 構成要素、およびサーバアダプタAPIを含む。該Java BeanおよびJava Script 構成要素は、Visual Java Scriptパレットから利用できるビルディングブロックの役割を果たす。

#### 【0225】

すべてのワークフローシステムツールは、Java Script 構成用紙だけではなく再利用可能なビーンズからも組み立てられる。該構成要素は、必要に応じて使用することができ、高いクライアント拡張性および顧客対応を可能にする。「Adapter Object (アダプタオブジェクト)」APIは、サーバ側で利用できる。サーバアダプタAPIによって、サードパーティは、該ワークフローエンジンを該所望の(DMS)、Directory (ディレクトリ) サービスおよびScript (スクリプト) インタプリタに接続できる。好ましい実施の態様においては、該APIは、該OMGのWorkflow Management Facility (ワークフロー管理機構) 基準をサポートする。サーバ側スクリプト機能は、機能性を追加構成要素/アプリケーションに拡張する。

#### 【0226】

該SDKの一つの態様とは、それがモデルビュー分離を実現するという点である。該モデルは、複数の同時プレゼンテーションを処理する。ユーザが該プレゼンテーションの一つで何かを変更すると、それが該モデルを変更し、それが、それから該プレゼンテーションの残りに通知する。Java Beans と呼ばれている該モデルクラスは、スクリプト機能に役立つ。それらをJava Beans として有することにより、それらは該SDKの別の態様である構成要素として使用できる。該SDKの別の態様とは、それがCORBA表現法を非表示にし、隠

すという点である。利点とは、該ワークフローシステムが、該ワークフローシステムモデルを使用している人々を、該クライアントと該サーバ間の該トランスポート層に関する表現法から遮るという点である。

【0227】

該SDKに加えて、該ワークフローシステムは、異なるクライアントを提供する。該開発者クライアントは、該開発者クライアントなしでは、該開発者はプロセスを設計することができないだろうため必須である。該開発者クライアントは該SDKの一部ではないが、該SDKは、該開発者クライアントがプロセス定義に入力を提供するため、該開発者クライアントなしにはほとんど役立たない。該開発者クライアントの意義とは、それが該開発者に該プロセス定義を入力させるという点である。該プロセス定義の入力の一部として、該開発者クライアントは該書式を作成するのを助ける。

【0228】

該開発者クライアントは、ビルディングブロック構成要素を使用して構築される。該開発者クライアントの内側には、開発者が書式を作成するのを助け、開発者が該サーバに該プロセス定義も入力させる該プロセス設計者ツールである。該SDK構成要素は、その他のクライアントだけではなく、該開発者クライアントを構築するために使用される。該開発者クライアントは、その他のクライアントとは対照的に、該エンジンに対する該プロセス定義のための入力を提供し、該書式の作成を助ける。アプリケーションは、任意のプロセス設計とは無関係に作成することができるが、有意義なシステムが該開発プロセスの最後に所望される場合には、該開発者は、該開発者クライアントを介してプロセス定義を入力できる。

【0229】

該ワークフローシステムには、該開発者クライアントが構築される同じ構成要素から構築される管理クライアントもある。まさに、該開発者クライアントが必要とされるように、該ワークフローシステムは、該管理クライアントが完全に機能的であることを必要とする。これは、該ワークフローシステムが、必ずしも、システムが完全に機能的であることを必要としない単純クライアントおよび具体

化クライアントとは対照的である。

#### 【0230】

該インタフェースおよびインプリメンテーションを提供することに加えて、該SDKは骨組を提供する。好ましい実施の態様においては、該骨組はJavaだけではなくC++でも提供される。このようにして、開発者は、専用のアダプタを開発するためにCORBA開発ツールを有する必要はない。

#### 【0231】

図17は、アプリケーション14、該ソフトウェア開発キット600の要素、および該(DMS)320およびDSオブジェクト360間の対話を示す。該SDKは、該WF Manager Proxy (WFマネージャ代理) 68、該Doc Repository (Docリポジトリ) 300、該(DMS) Implementation ( (DMS) インプリメンテーション) 310、該FTPユーティリティ350、および該Directory Service (ディレクトリサービス) 24から構成されている。

#### 【0232】

該ワークフローシステムは、任意の種類のクライアントを形成するために該ビルディングブロックを組み合わせることを可能にするビルディングブロック構成要素を有する。その結果、クライアントは、柔軟に置換可能な構成要素を有し、それによりワークフロークライアントアプリケーションを作成する上で該開発者を可能にする。該ワークフローシステムクライアント構成要素は、Visual Java Scriptなどのビルダツールを使用して統合され、カスタマイズされるのに向いている。

#### 【0233】

クライアントモデル構成要素は、非対話方Javaビーンズである。該クライアントモデル構成要素の目的とは、I I O Pを介した該サーバとのその対話だけではなく、クライアントオブジェクトの状態もカプセル化することである。

#### 【0234】

さまざまなタイプのユーザ向けの複数のクライアントユーザインタフェースがある。該クライアントUI構成要素の目的とは、作業リスト一覧表示ユーザイン

タフェースなどの該ワークフローシステムのユーザインタフェースのさまざまな構成要素を提供することである。これらの構成要素は、クライアントオブジェクトを維持したり、該ワークフローシステムサーバとの対話を実行しないことになる。

#### 【0235】

該APIは、該ワークフローシステムサーバに対するすべてのインタフェースをカバーする該モデルビーンズを含むJavaパッケージとして提供される。このAPIによって、該開発者は、その管理を含む、該ワークフローシステムのすべての態様にアクセスし、操作することができる。

#### 【0236】

開放アダプタAPIによって、サードパーティが、該ワークフローシステムエンジンを該所望される(DMS)またはDSに接続できるようにする。DSアダプタは、すでに、NT User Registry (NTユーザレジストリ)、MS Exchange (MS交換機)、およびLDAPのために提供される。(DMS)アダプタは、該NT File System (NTファイルシステム)、MS Exchange (MS交換機)、およびJDKのために提供される。該開放アダプタAPIは、それ以外のシステムと動作するアダプタを開発することができるように提供される。アダプタインタフェースごとの該CORBA IDL仕様に、例のコードだけではなく該ワークフローソフトウェアパッケージが備えられる。

#### 【0237】

該Adapter API (アダプタAPI) は、さまざまな言語の中に文字を収容するために一意の文字を表すための幅広い文字列を活用する。8-ビット文字はラテン文字およびアラビア文字を表すには十分であるが、それ以外の言語の文字はそれらを表すためにさらに多くのビットを必要とする。漢字は、少なくとも二重バイトを必要とする。

#### 【0238】

該Client API (クライアントAPI) は、該ワークフローシステムサーバへのすべてのインタフェースをカバーするモデルクラスを含むJavaパ

パッケージとして提供される。該Client API（クライアントAPI）によって、開発者は、その管理を含む、該ワークフローシステムのすべての態様にアクセスし、操作することができる。クライアントビーンAPIは、該開発者が専用のアプリケーション/タスクを構築するために、該開発者によって使用できる。該APIによって、開発者は、サードパーティの製品およびツールと該ワークフローシステムを統合するだけでなく、ワークフローシステム機能性をカスタマイズし、拡張することもできる。

#### 【0239】

Client（クライアント）ビーンAPIがアプリケーションを構築するために使用されると、該アプリケーションの最重点は、該ワークフローマネージャパッケージである。該ワークフローマネージャパッケージは、該Server（サーバ）に対する接続を確立し、サーバ側オブジェクトへの参照を検索するためにビーンを含む。Client（クライアント）ビーンAPIは、該ワークフローサーバへログインする、作業アイテム、プロセス、またはプロセス定義（テンプレート）を検索する、該Document Management System（文書管理システム）を管理する、および役割メンバーシップを決定するなどの共通のタスクを実行するために使用できる。すべてのワークフローシステムツールは、JavaScript構成要素だけではなく、これらのクラスからも開発されてよい。

#### 【0240】

すべての種類のワークフローアプリケーションは、該モデルクラスから開発できる。該ユーザは、該クラスを使用して、Javaアプリケーションを開発することができるか、あるいは該ユーザは該クラスでの該ブラウザで利用可能な該スクリプト機能を行ってよい、すなわち該クラスで利用可能なブラウザスクリプト機能を使用し、Javaアプリケーションを開発する。

#### 【0241】

表5は、該ワークフローシステムClient（クライアント）のパッケージの要約を含む。これらのパッケージは、該開発者が利用できるようになったクラスおよびインタフェースを含む。パッケージごとに該完全に限定された名前は、

COM. fujitsu. fsc. TWFlow. Client. <package name>である。

【0242】

該Process（プロセス）変更制御機構は、変更を、同じ計画を実行しているすべてのプロセスに適用し、有意義な通知を関係しているユーザに提供する。該プロセス変更制御機構は、ユーザに、指定された計画を実行しているすべてのプロセスを一時停止する、該計画を変更する、および該ユーザによって入力された該データを救出する一方で、該修正された計画を実行しているすべてのプロセスに対し変更（変形）を適用する能力を与える。

【0243】

カレントオーガナイザに対する代替策として、本発明の実施の態様は、フォルダのツリーであるフォルダ階層を提供してよい。該フォルダのツリーは、パーソナルフォルダサブツリー、および共用されているフォルダサブツリーを含む。該パーソナルフォルダサブツリーは、割り当てられたタスクのためのフォルダ（導出済み／計算済み）、所有されているプロセス（導出済み／計算済み）、およびOthers（その他）を含む。該割り当てられたタスクのためのフォルダは、アクティブであるか、あるいは完了していることがある。該所有されているプロセスは、アクティブであるか、あるいは完了していることがある。その他は、ユーザが、プロセス、計画、および活動に対するショートカットを作成できるパーソナルフォルダ階層を作成できるようにする。該ワークフローシステム管理者によって管理される該Shared（共用）フォルダサブツリーは、発行済みの計画を含む。それは、該（DMS）にサブツリーを提供し、サーバにサブツリーを提供する。

【0244】

導出済み／計算済みフォルダは、（リレーショナルデータベース照会に類似する）フィルタを使用して実行時に計算される内容を有する。ユーザによる該フォルダの要求時に、フィルタ（照会）が評価され、該導出済み／計算済み結果は該ユーザに送信される。指定されたフォルダに対する該初期要求の後、該フィルタによって指定される該選択基準を満たす該ワークフローシステムでのあらゆる変



更は、先を見越して該要求者に伝搬される。

【0245】

本発明の別の実施の態様においては、プロセスは単一レベルで表示され、該ユーザはそれらを並べ替え、フィルタリングすることができる。本発明のやはり別の実施の態様においては、該フォルダ組織は結合的かつ多次元的であり、フィルタは該フォルダに適用することができる。該オーガナイズは、図11の左上隅に示されている。該活動、プロセス、テンプレートタブに加えて、該ユーザは、該リストがどのように表示されるのか、すなわちフィルタおよび並べ替えを制御する二つの制御機構を有する。使用されている該パラダイムは、結合的なリストに基づいている。ユーザは、ほぼ構造化検索言語（SQL）のようにフィルタおよび並べ替え順序を指定する。好ましい実施の態様においては、該ワークフローシステムは、プロセス用の以下のユーザによって提供されているフィルタを有する。

【0246】

すなわち、MYactive、MYInactive、ALLactive、ALLinactive、およびEverythingである。MYactiveは、すべてのアクティブユーザプロセスのリストを提供する。MYInactiveは、すべてのイナクティブユーザプロセスのリストを提供する。ALLactiveは、それらがユーザのプロセスであるかどうかに関係なく、すべてのアクティブプロセスのリストを提供する。ALLinactiveは、それらがユーザのプロセスであるかどうかに関係なく、すべてのイナクティブプロセスのリストを提供する。Everythingは、すべてのプロセスのリストを、アクティブであっても、イナクティブであっても提供する。

【0247】

本発明の好ましい実施の態様においては、該ワークフローシステムを使用すると、ユーザは、先を見越した通知を用いて、あるいは先を見越した通知を使用しないで該リストを要求することができ、該ユーザが知りたいと考える新しいプロセスが作成される場合には、該先を見越した通知を用いると該リストが更新されるが、先を見越した通知を使用しないと、該リストは更新されない。

## 【0248】

該書式記入フィールドは、プロセス設計者によって生成され、Eメール、アドレス、題名等の該ユーザプロフィールの一部であるフィールドの事前に定義されたセットである。

## 【0249】

該ワークフローシステムは、役割分解およびメール通知を、匿名ユーザ（該ディレクトリ内にアカウントを持たないユーザ）に提供し、特定し、Eメールを介した活動要求を通知することができるだけのユーザの参加を可能にする。この場合、役割メンバーシップの概念は、ユーザ識別としてEメールを含むように拡大されなければならない。





## 【0250】

## 【表1】

メニュー	メニューコマンド	説明
Flow (フロー)	Login (ログイン)	ワークフローシステム Server (サーバ) に接続するために使用する。
	Logout (ログアウト)	該ワークフローシステム Server (サーバ) から切断するために使用する。
	User Profile (ユーザプロフィール)	ユーザの個人的なユーザ設定値を設定するためのダイアログボックスを表示するために使用する。
View (ビュー)	Show Graphical View (グラフィックビュー表示)	選択された活動が属する該ワークフローを表示するために使用する。
	Show Detailed View (詳細ビュー表示)	プロセス内の現在の活動の詳細を表示するために使用する。
	Widen Canvas (キャンバス拡大)	該 Target Area (ターゲット領域) がワークフローを表示するときに、可視 Target Area (ターゲット領域) を拡大するために使用する。
	Heighten Canvas (キャンバス強調表示)	該 Target Area (ターゲット領域) がワークフローを表示するときに、可視 Target Area (ターゲット領域) を強調表示するために使用する。
Process (プロセス)	New (新規)	選択されたテンプレートに基づき新規プロセスを作成するために使用する。
	Properties (プロパティ)	選択されたプロセスのプロパティ (General (一般)、Initiator (開始プログラム)、User Defined Attributes (ユーザ定義属性)、および Attachments (添付物)) を表示するために使用する。
	History (履歴)	プロセスの履歴を表示する。
Activity (活動)	Properties (プロパティ)	選択されたプロセスのプロパティ (General (一般)、Assignee (譲受人)、User Defined Attributes (ユーザ定義属性)、Scripts (スクリプト) および Forms (書式)) を表示するために使用する。
Template (テンプレート)	New (新規)	新規テンプレートを作成するために使用する。
	Edit (編集)	既存のテンプレートを編集するために使用する。
	Save (保存)	テンプレートを保存するために使用する。
	Save As (として保存)	別の名前で既存のテンプレートを保存するために使用する。
	Properties (プロパティ)	選択されたプロセスのプロパティ (General (一般)、Owner (オーナー)、User Defined Attributes (ユーザ定義属性)、および Script (スクリプト)) を表示するために使用する。
Document (文書)	Associations (関連付け)	指定されたファイル名拡張子の付いた添付物を開くためにどのアプリケーションを使用するのかを指定するために使用する。
	Check-in (返却)	他の人が変更を見て、必要に応じてそれらを編集できるように、編集のために開かれた添付物を返すために使用する。
Help (ヘルプ)	Help Topics (ヘルプトピック)	オンラインヘルプを表示するために使用する。






[0251]

【表2】

	新規プロセスを作成するために使用する。
	Graphical View (グラフィックビュー) にトグルするために使用する。
	Detailed View (詳細ビュー) にトグルするために使用する。
	ユーザの User Profile (ユーザプロファイル) を表示するために使用する。



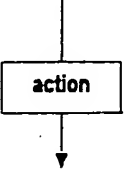
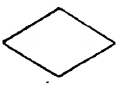
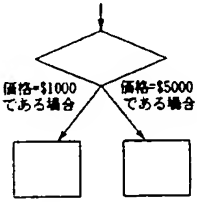
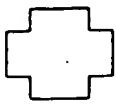
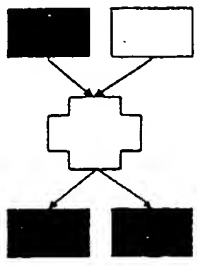
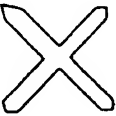
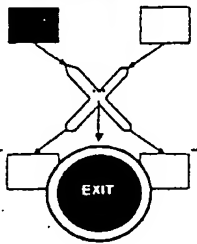


【0252】

【表3】

	ユーザが開始したアクティブプロセスに印を付ける。
	ユーザが開始した完了したプロセスに印を付ける。
	アクティブプロセスに印を付ける。
	完了したプロセスに印を付ける。
	テンプレートを示す。

【0253】

【表4】

演算子	記号	説明	使用法図
Start (開始)		プロセスの始まりを特定する。あらゆるプロセスは1つのおよび1つだけのStart (開始) ノードを持たなければならない。	
Arrow (矢印)		イベントのフローを定義する。矢印が活動ノードから出ると、それは、該活動譲受人が、該活動に応じて行うことができる選択を表す。	他の使用法の図を参照すること。 注記：すべての矢印には名前フィールドがあるが、このフィールドは使用法図から省略されている。
Condition (条件)		このノードに到達すると、該プロセスは、該ノードの出矢印の1つによって指定される条件が満たされるまで続行しないだろう。この種のノードを使用して、ユーザにとっての自動化意思決定を行う。	
OR (論理和)		この種のノードに先行し、この種のノードに繋がる第1活動が完了すると、該ノードに繋がれるすべての付随する活動が同時に起動される。複数の新しい活動を生じさせるために、活動に対する応答を必要とするときには、この種のノードを使用すること。	
AND (論理積)		このノードに達すると、プロセスは、このノードにつながるすべての活動が完了されるまで続行しないだろう。活動を同期するには、このノードを使用すること。	
Exit (終了)		プロセスの最後を特定する。あらゆるプロセスは、少なくとも1つのExit (終了) ノードを持たなければならない。	
記号一覧 (緑) = 現在の活動、(青) = 完了した活動、(明るい緑み青) = 未決活動			

【0254】

【表5】

パッケージ名	説明
appinterfa ces	3つの主要なUI構成要素内のデータを更新するため、および編集セッションを開始するためのインタフェースのセットを含む。
(DMS)	Document Management System (文書管理システム) ((DMS)) および Directory Service (ディレクトリサービス) (DS) を管理するためのインタフェースおよびクラスを含む。これは、該(DMS)内のフォルダと添付物に関する、または該DS内の役割とグループに関する情報の検索/更新、ならびに該(DMS)からのオブジェクトの返却/貸出しを含む。
event (イベント)	ユーザが新しいアイテムを作成するか、それに関して情報を取得する別のアイテムを選択するときに、傾聴するためのインタフェースおよびクラスを含む。
form (形式)	形式から情報を設定、および取得するために使用されるアプレットクラスを含む。
Organizer 2	Organizer (オーガナイザ) UI構成要素内のイベントを傾聴し、該Organizer (オーガナイザ) を適切に更新するためのインタフェースおよびクラスを含む。
plan (計画)	計画またはプロセスモデルのすべての異なる態様を更新するためのインタフェースおよびクラスを含む。これは、ノードと矢印、書式と添付物、作業アイテム、および許可レベルを含む。
planview	appinterfaceパッケージ内でインタフェースを実現するために使用されるクラスを含む。
WFManager	Server (サーバ) に対する初期接続を取得し、フィルタ基準に基づく作業アイテム、プロセス、およびテンプレートリストを検索し、プロセスおよびテンプレートを検索するために使用されるクラスを含む。

## 【0255】

ここに、本発明の好ましいおよび例示的な実施の態様であると考えられるものが説明されてきたが、本発明のそれ以外の修正は、ここの教示から当業者にとっては明らかなものとし、したがって、添付クレームにおいて、すべてのこのような修正および拡大が、真の本発明の精神および範囲に該当することが確実であることが所望される。本発明は、そのすべての実施の態様を含むと解釈されなければならない。加えて、当業者は、他のアプリケーションが、本発明の精神および範囲から逸脱することなく、ここに述べられるものの代わりに使用されてよいことを容易に理解することができる。

## 【図面の簡単な説明】

## 【図1】

総合的な分散ワークフローシステム構成要素アーキテクチャおよび本発明の実施の態様の構成要素対話を示す図である。

## 【図2】

骨組みおよびスタンプを作成することに関わる多様な要素を示すブロック図である。

## 【図3】

DD - (DMS) - (DS) 構造を示す図である。

## 【図4】

クライアントと (DMS) 要素の間の対話を示す図である。

## 【図5】

単一サーバ構成のワークフローシステムアーキテクチャを示す図である。

## 【図6】

ウェブクライアントのデフォルト構造を示す図である。

## 【図7】

ワークフローシステムスクリプトフレームワークを示すブロック図である。

## 【図8】

ウェブベースのDeveloper Client (開発者クライアント) の画面表示を示す図である。

## 【図9】

該Developer Clientの画面レイアウトの画面領域および名称を示す図である。

## 【図10】

汎用活動リストを示す図である。

## 【図11】

画面表示の右側フレーム内のワークフロープロセスを示す図である。

## 【図12】

活動の図式による表記を示す図である。

## 【図13】

ノードおよび矢印を配置するために使用されるTarget Area Toolbar (ターゲット領域ツールバー) を示す図である。

## 【図14】

ウェブベースのSimple (単純) クライアントの画面表示を示す図である。

## 【図15】

二つの要素フレームとしての該Simple Client (単純クライアント) ビューを示す図である。

## 【図16】

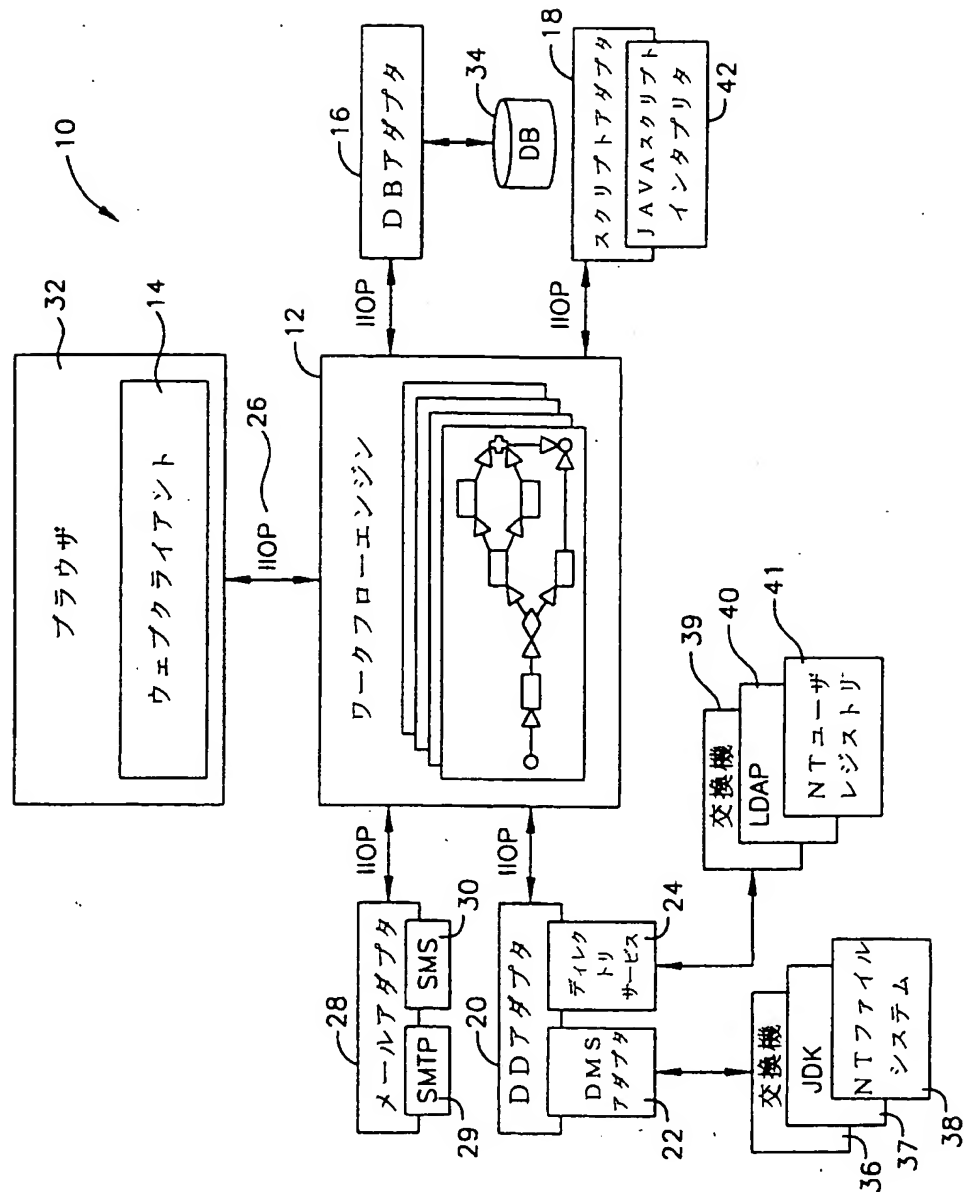
Eメールクライアントインタフェースを示す図である。

## 【図17】

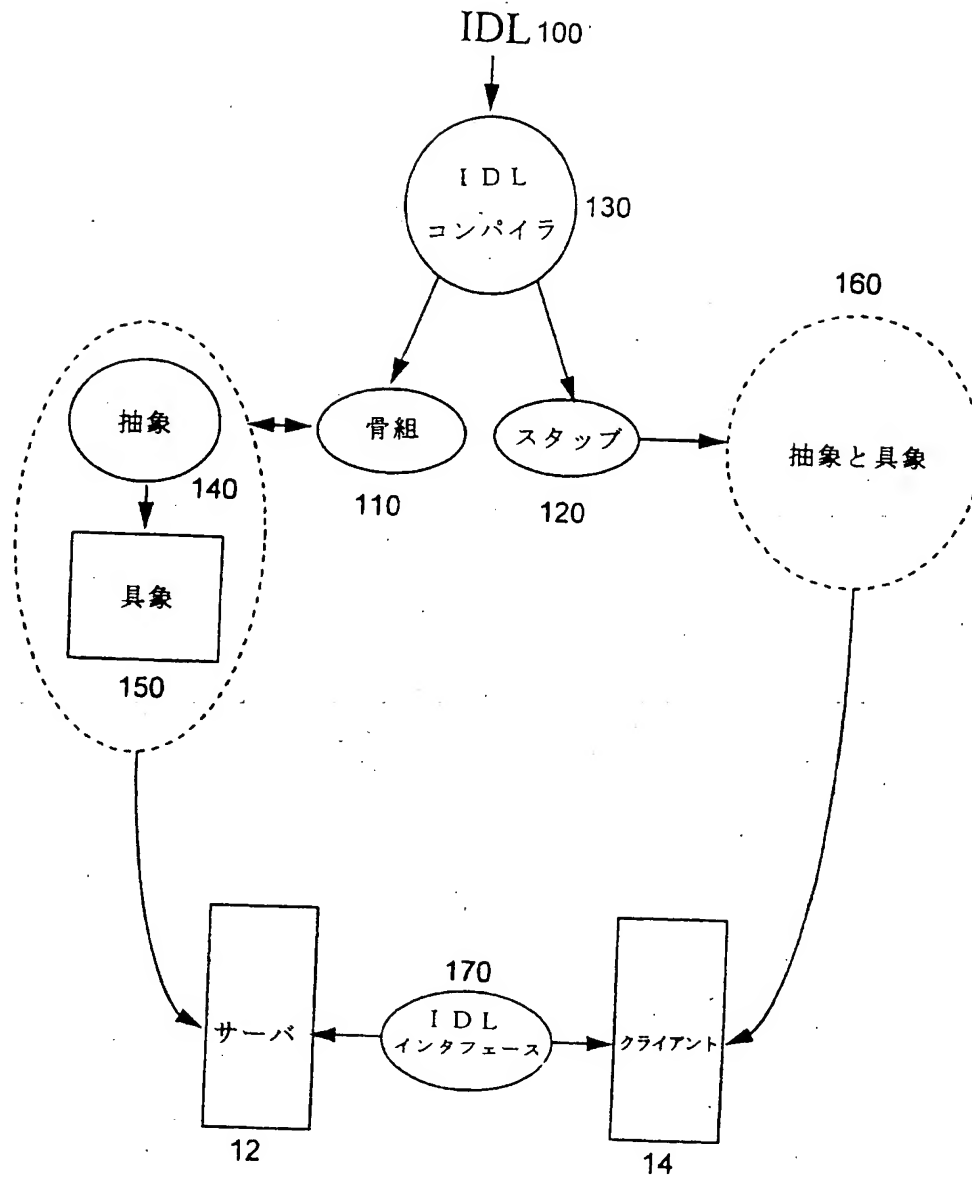
アプリケーション、ソフトウェア開発キットの要素および(DMS) オブジェクトと(DS) オブジェクトの間の対話を示す図である。



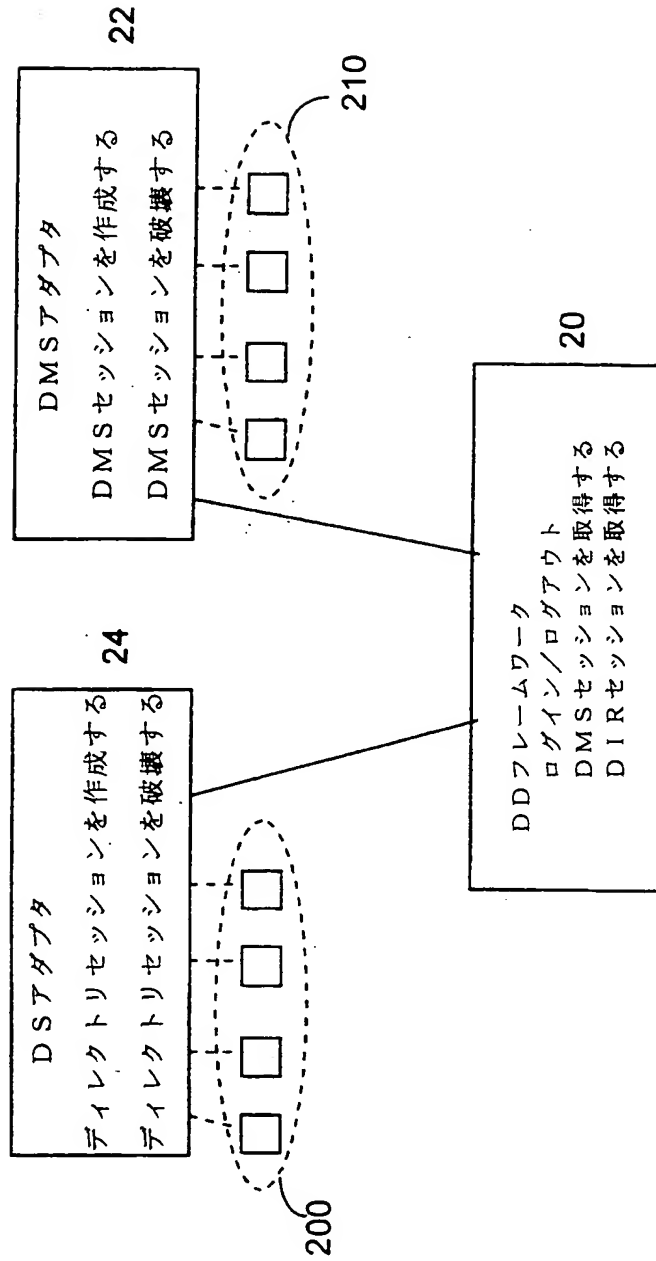
【図1】



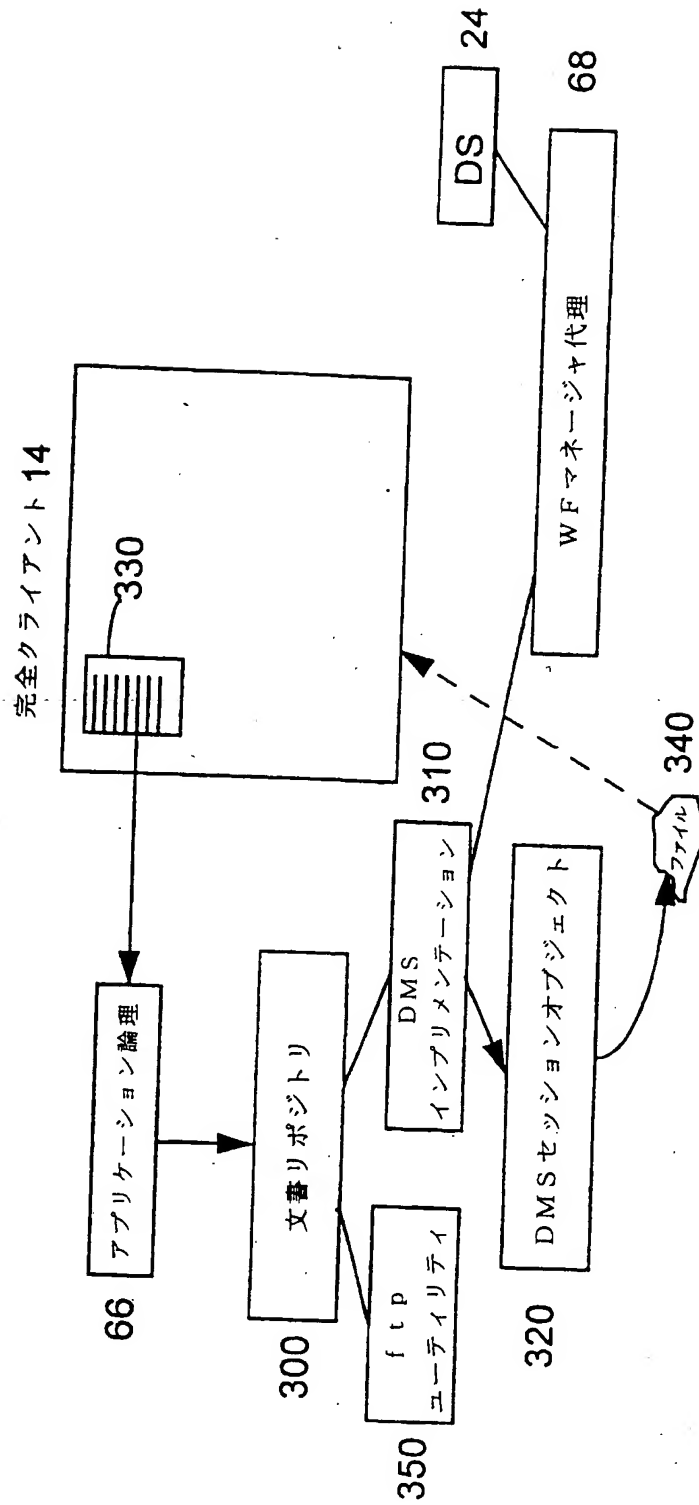
【図2】



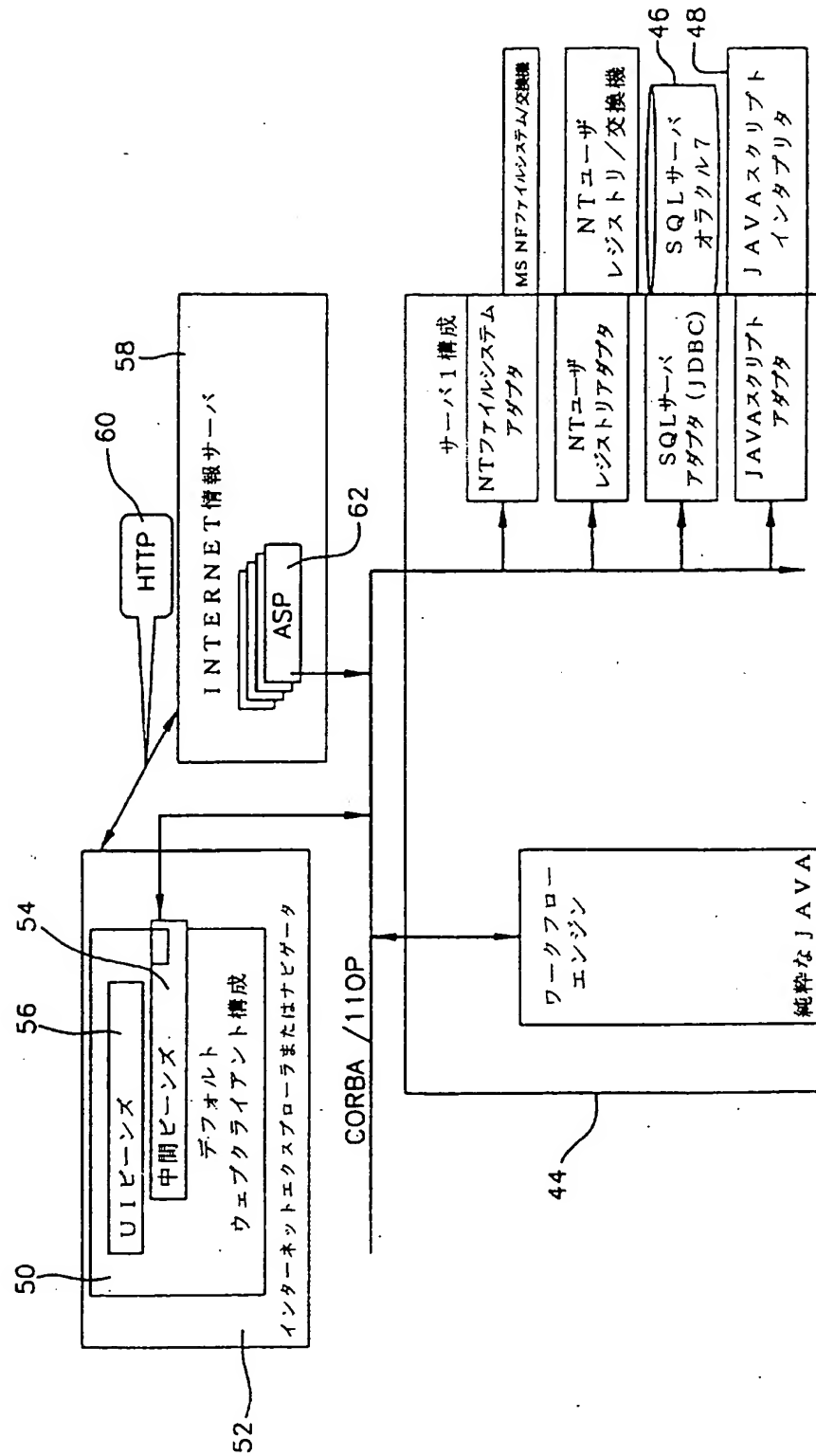
【図3】



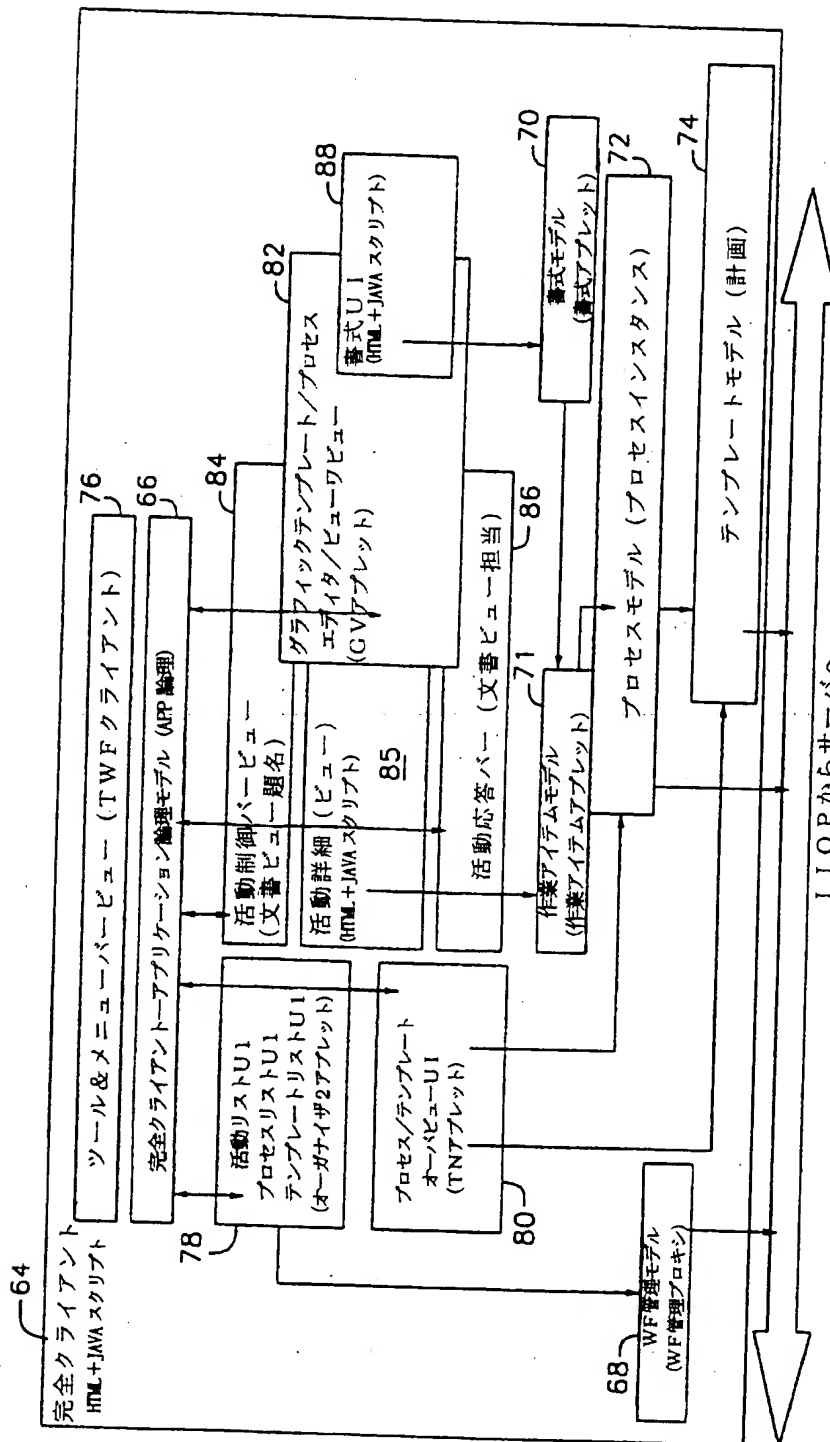
【図4】



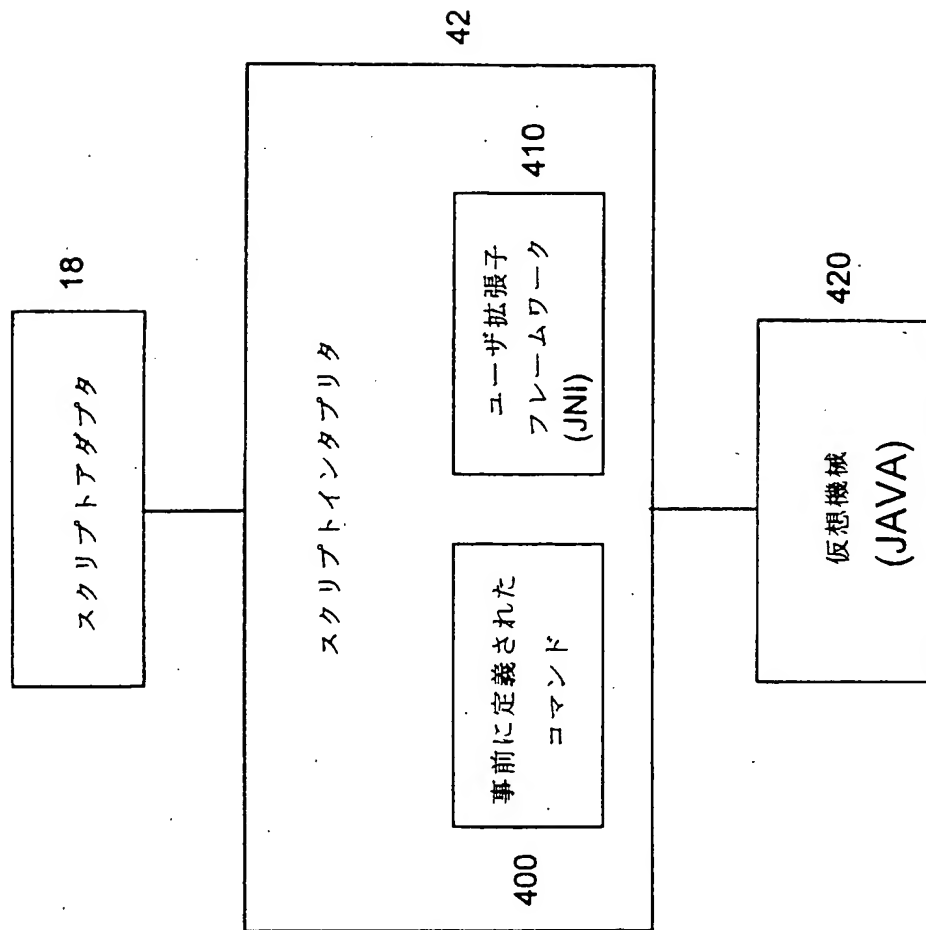
【図5】



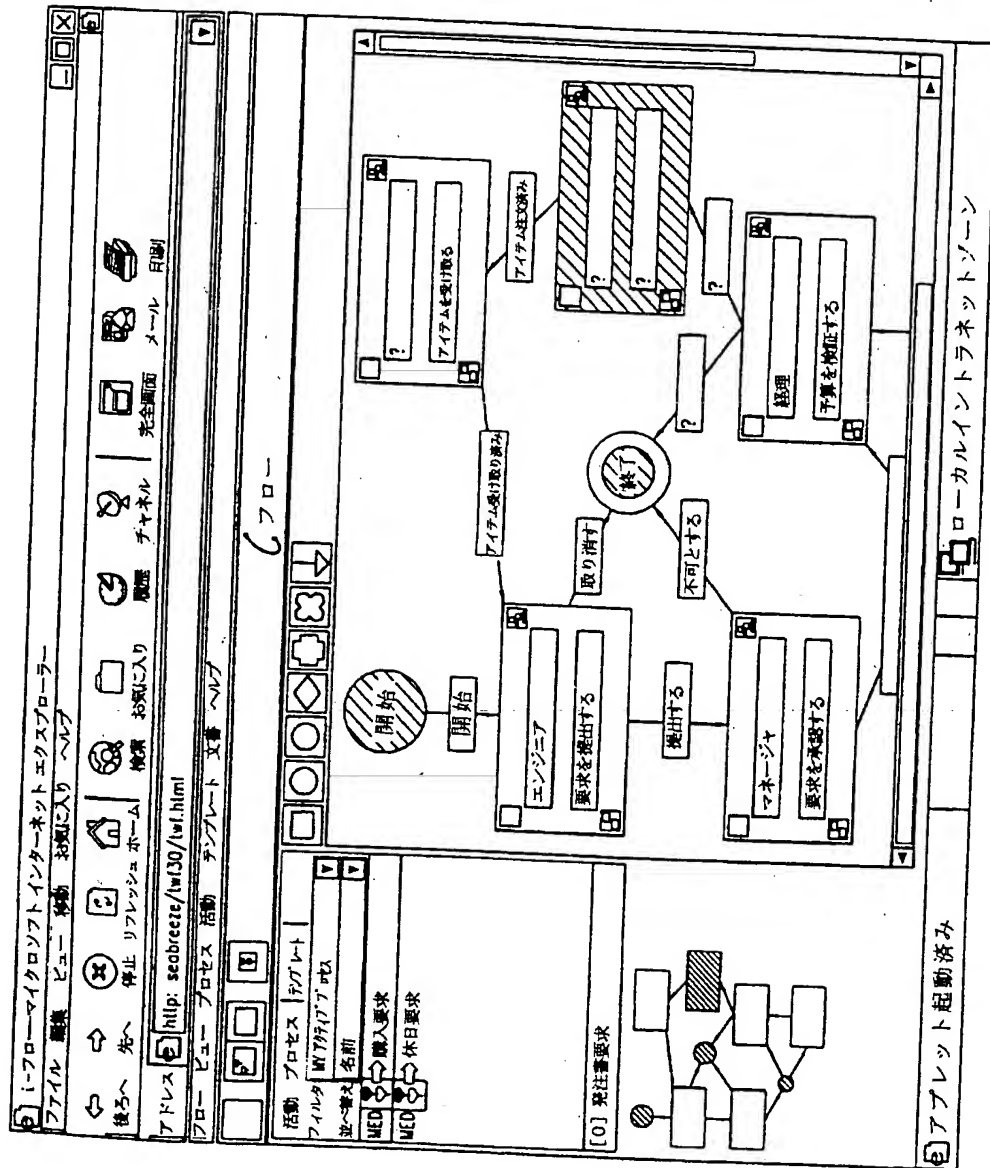
【図6】



【図7】

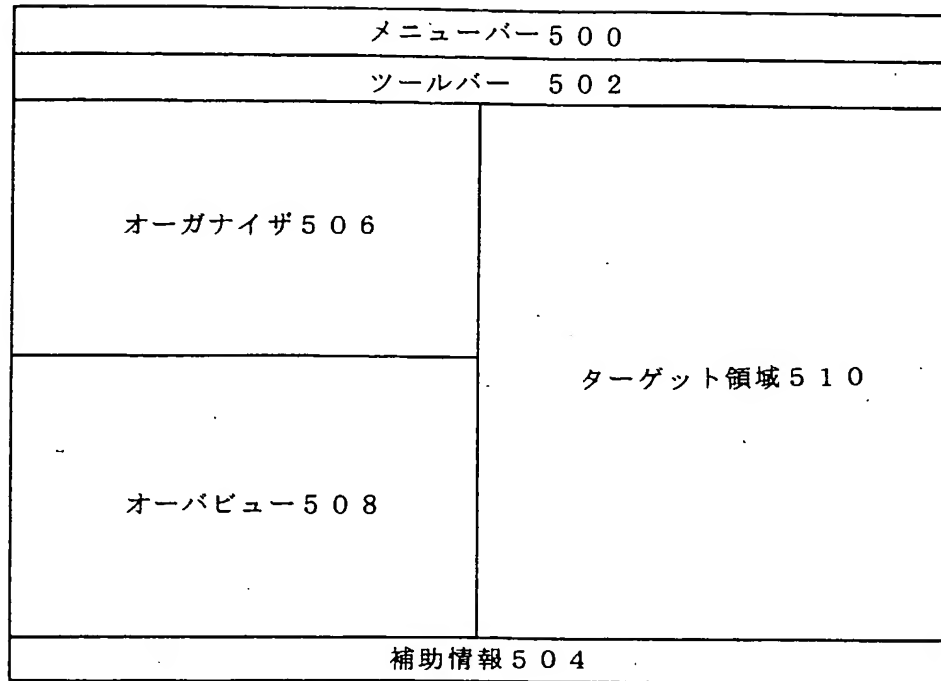


[ 8 ]

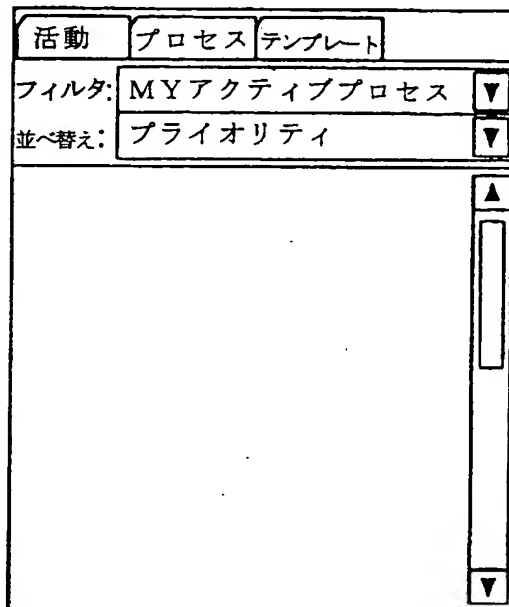




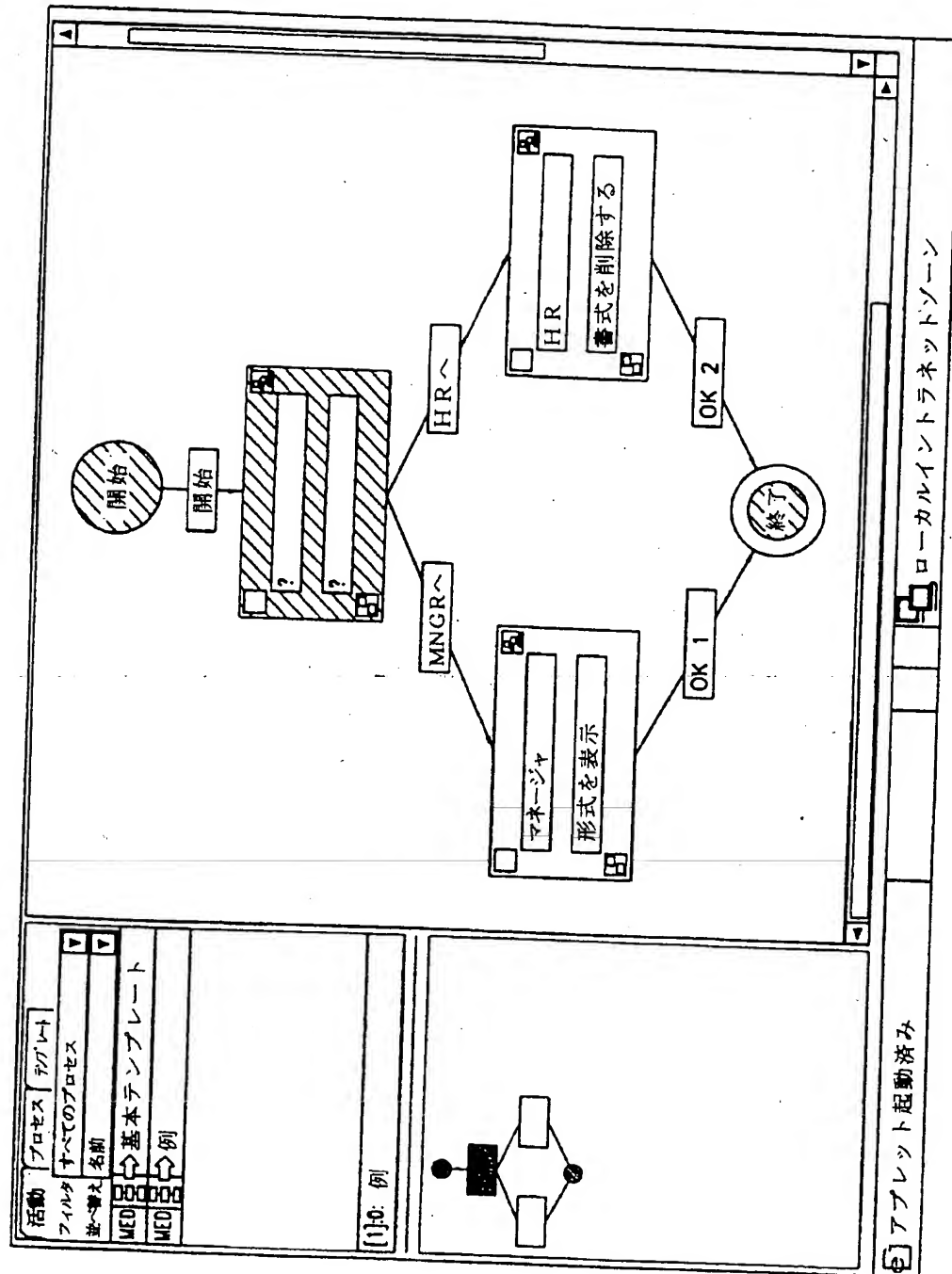
【図9】



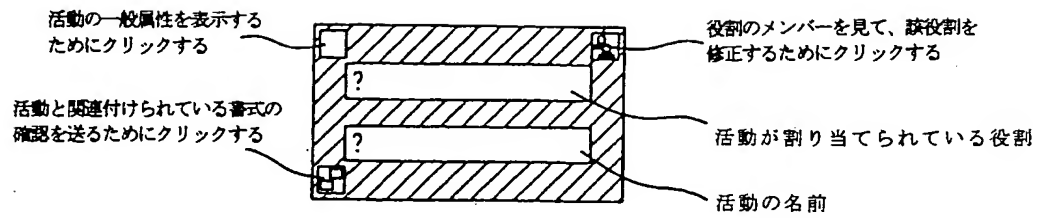
【図10】



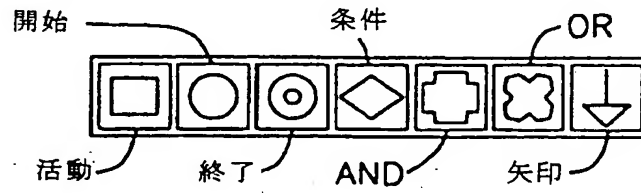
【図11】



【図12】



【図13】



【図14】

インターネットエクスプローラ  
ファイル 編集 ビュー 移動 お気に入り ヘルプ  
戻るへ 先へ 停止 リフレッシュ ホーム 検索 お気に入り 履歴 チャンネル 完全画面 メール 印刷  
アドレス [http://mars/w130/wfsimple\\_en.html](http://mars/w130/wfsimple_en.html)

アクティビティデスクトップ  
フロー単純クライアント  
新規プロセス 追加物追加 関連付け

作業アイテム

添付物

プロセス: BUDGET提出プロセス  
活動: 活動名  
日付: '69.12.31. 4:00:00PM  
宛先: WEIN  
送り主: MIKE

添付物名  
1 活動名  
COMPANY POLICY. DOC  
Priceブック. PDF  
カスタマ要件. DOC  
予算. XLS  
書式  
書式名  
プロジェクト予算  
人員予想  
リソース予想

記述: ユーザの予算要件を提出せよ。

アクション2

ローカルイントラネットゾーン

【図15】

作業リスト	E-メールクライアントビュー (HTML * JavaScript)
-------	---------------------------------------

【図16】

1-フロー-マイクロソフトインターネットエクスプローラ  
 ファイル 編集 ビュー 移動 お気に入り ヘルプ  
 戻るへ 先へ 停止 リフレッシュ ホーム 検索 お気に入り 履歴 チャネル 完全画面 メール 印刷  
 アドレス [http://mars/tw/30/twemail\\_en.html?user=wein&id=2&server=MARSflow.30](http://mars/tw/30/twemail_en.html?user=wein&id=2&server=MARSflow.30)

フローEメールクライアント Eメールデスクトップ

追加物追加 関連付け

プロセス: BUDGET提出プロセス  
 活動: 活動名  
 日付: '69.12.31. 4:00:00PM  
 宛先: WEIN  
 送り主: MIKE

添付物  
 添付物名  
 COMPANY POLICY.DOC  
 Priceブック.PDF  
 カスタマ要件.DOC  
 予算.XLS

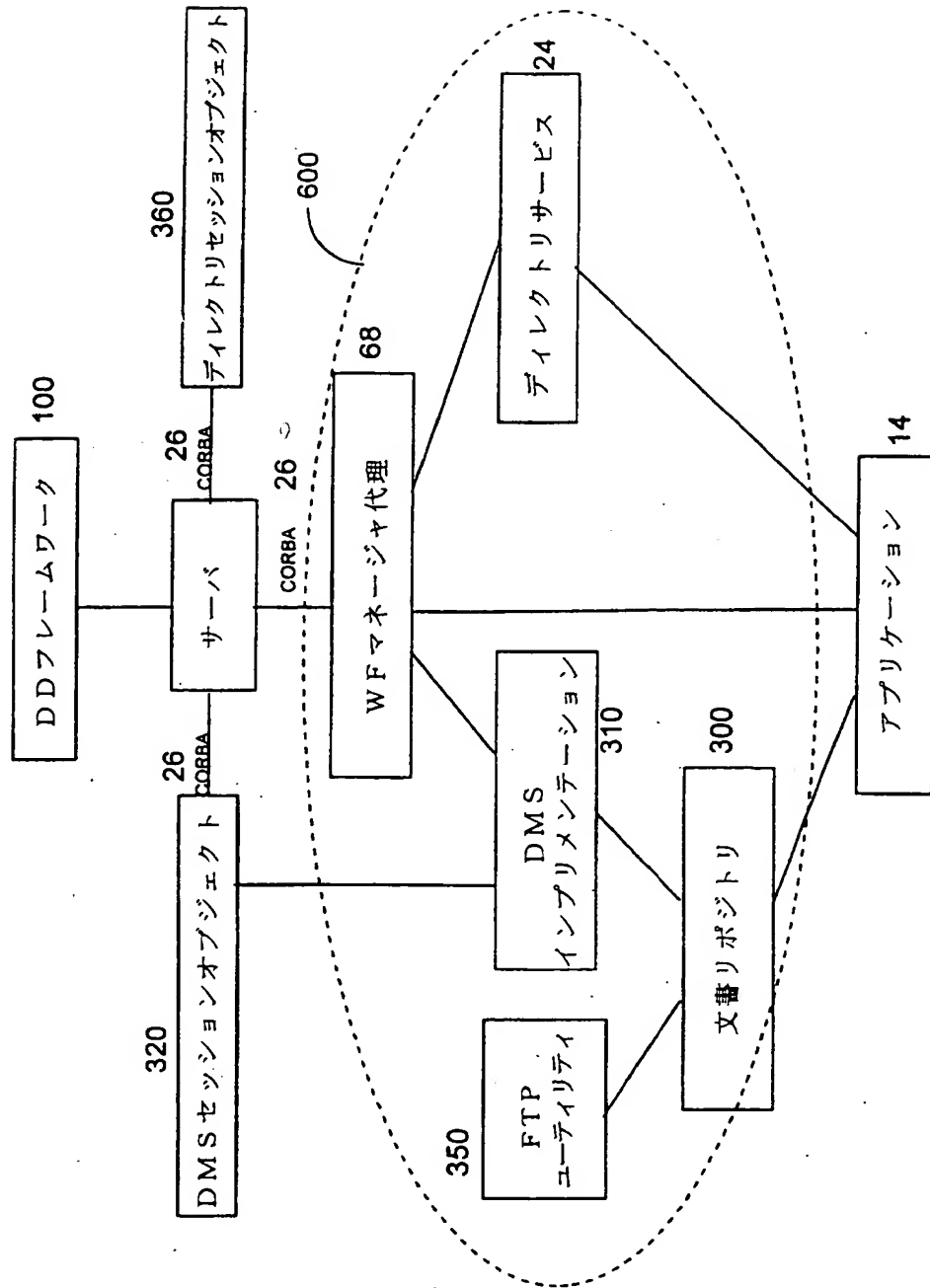
書式  
 書式名  
 プロジェクト予算  
 人員予想  
 リソース予想

記述: ユーザの予算要件を提出せよ。

アクション2

実行済み ローカルイントラネットゾーン

【図17】



## 【国際調査報告】

## INTERNATIONAL SEARCH REPORT

International application No.  
PCT/US99/19232

<b>A. CLASSIFICATION OF SUBJECT MATTER</b> IPC(7) : G06F 17/00, 17/60 US CL : 707/103 705/8 According to International Patent Classification (IPC) or to both national classification and IPC		
<b>B. FIELDS SEARCHED</b> Minimum documentation searched (classification system followed by classification symbols) U.S. : 707/103 705/8 Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) IEEE, EAST		
<b>C. DOCUMENTS CONSIDERED TO BE RELEVANT</b>		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5,721,913 (ACKROFF et al.) 24 February 1998, abstract, fig. 1, 3, col. 1, line 16 to col. 2, line 54.	1-59
X	US 5,826,239 (DU et al.) 20 October 1998, abstract, fig. 2, 4-6, col. 1-4	1-59
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> See patent family annex.		
Special categories of cited documents: * "A" documents defining the general state of the art which is not considered to be of particular relevance * "B" earlier documents published on or after the international filing date * "L" documents which may throw doubts on priority claim(s) or which is cited to establish the publication date of another claim or other special reason (as specified) * "O" documents referring to an oral disclosure, use, exhibition or other means * "P" documents published prior to the international filing date but later than the priority date claimed * "T" later documents published after the international filing date or priority date and not in conflict with the application but cited to understand the principle of theory underlying the invention * "X" document of particular relevance; the claimed invention cannot be considered novel or obvious to involve an inventive step when the document is taken alone * "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art * "Z" document recader of the same patent family		
Date of the actual completion of the international search 21 MAY 2000		Date of mailing of the international search report JAN 19 2001
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230		Authorized officer KEVIN TESKA James R. Matthews Telephone No. (703) 305-9704

Form PCT/ISA/210 (second sheet) (July 1998)\*



## フロントページの続き

- (31) 優先権主張番号 60/097,791  
(32) 優先日 平成10年8月25日(1998. 8. 25)  
(33) 優先権主張国 米国(US)  
(81) 指定国 JP, US  
(72) 発明者 サンディーブ ティワリ  
アメリカ合衆国, カリフォルニア州  
95032, ロスゲイトス, モジョニーラ コ  
ート 103  
(72) 発明者 ジェリー ウェイン  
アメリカ合衆国, カリフォルニア州  
95125, サンホセ, フェアーヒル レイン  
2272  
(72) 発明者 アンドリュー ジュン  
アメリカ合衆国, カリフォルニア州  
95117, サンホセ, ミツィ ドライブ  
4184  
Fターム(参考) 5B085 AA01 AA08 BE03

**THIS PAGE BLANK (USPTO)**